NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

FIDUCIAL MARKER DETECTION AND POSE ESTIMATION FROM LIDAR RANGE DATA

by

Richard Morrison

March 2010

Thesis Advisor:	Mathias Kölsch
Co-Advisor:	Timothy H. Chung

This thesis was done at the MOVES Institute
Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2010	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Fiducial Marker Detection and Pose Estimation From LIDAR Range Data **6. AUTHOR(S)** Richard B. Morrison	**5. FUNDING NUMBERS**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Light Detection and Ranging (LIDAR) systems are three dimensional (3D) imaging sensors applied for mapping terrain, measuring structural dimensions, and navigating robots. Pulsed laser rangefinders provide precise range measurements that require an estimate of sensor pose for transformation into world coordinates. Pose information is frequently provided with extrinsic sources such as Global Positioning System (GPS) or an Inertial Measurement Unit (IMU). Unreliable signal availability for GPS in military environments and the high cost of IMUs limit the employment of these extrinsic sources. Determining pose intrinsically by detecting landmarks in the environment within the sensor data is more ideal. Fiducial markers with known geometric dimensions and orientation provide a means of estimating LIDAR pose and registering data. Presented is a method for landmark detection and pose estimation within range data. Cylinder, cone, and sphere geometries are assessed for use as fiducial markers. The detection algorithm extracts geometric features from LIDAR point data and tests for fit to a fiducial marker model. Geometric feature extraction compresses the data set and leads to a potential intrinsic registration method using environment landmarks. The detection accuracy and pose estimation precision are examined with terrestrial LIDAR range data captured in various outdoor street environments.

14. SUBJECT TERMS LIDAR, point cloud, feature extraction, fiducial marker, pose estimation	**15. NUMBER OF PAGES** 121
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

FIDUCIAL MARKER DETECTION AND POSE ESTIMATION FROM LIDAR RANGE DATA

Richard B. Morrison
Lieutenant Commander, United States Navy
B.S., Worcester Polytechnic Institute, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)

from the

NAVAL POSTGRADUATE SCHOOL
March 2010

Author: Richard B. Morrison

Approved by: Mathias Kölsch
Thesis Advisor

Timothy H. Chung
Co-Advisor

Mathias Kölsch
Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Light Detection and Ranging (LIDAR) systems are three dimensional (3D) imaging sensors applied for mapping terrain, measuring structural dimensions, and navigating robots. Pulsed laser rangefinders provide precise range measurements that require an estimate of sensor pose for transformation into world coordinates. Pose information is frequently provided with extrinsic sources such as Global Positioning System (GPS) or an Inertial Measurement Unit (IMU). Unreliable signal availability for GPS in military environments and the high cost of IMUs limit the employment of these extrinsic sources. Determining pose intrinsically by detecting landmarks in the environment within the sensor data is more ideal. Fiducial markers with known geometric dimensions and orientation provide a means of estimating LIDAR pose and registering data. Presented is a method for landmark detection and pose estimation within range data. Cylinder, cone, and sphere geometries are assessed for use as fiducial markers. The detection algorithm extracts geometric features from LIDAR point data and tests for fit to a fiducial marker model. Geometric feature extraction compresses the data set and leads to a potential intrinsic registration method using environment landmarks. The detection accuracy and pose estimation precision are examined with terrestrial LIDAR range data captured in various outdoor street environments.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
ALS	Airborne LIDAR System
AMCW	Amplitude Modulated Continuous Wave
API	Application Programming Interface
CW	Clockwise
CCW	Counter Clockwise
FMCW	Frequency Modulated Continuous Wave
FOV	Field of View
FPA	Focal Plane Array
FPS	Frames Per Second
FRE	Fiducial Registration Error
GIS	Geographic Information Systems
GPS	Global Positioning System
Hz	Hertz, cycles per second
INS	Inertial Navigation System
IMU	Inertial Measurement Unit
LED	Light Emitting Diode
LIDAR	LIght Detection and Ranging
OSG	OpenSceneGraph
RANSAC	RANdom Sampling And Consensus
RMS	Root Mean Square
RPM	Revolutions Per Minute
SLAM	Simultaneous Location and Mapping
TIN	Triangular Irregular Network
TLS	Terrestrial LIDAR System
VLS	Velodyne LIDAR System

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

I would like to thank the many people that have helped me to complete this research. First is my wife, Carrie, for being so patient and supportive while I spent many hours in the classroom and in the lab away from home. I would also like to thank my thesis advisors, Dr. Mathias Kölsch and Dr. Timothy Chung for their enormous confidence in my abilities, their technical guidance, and their patient support. Without their inspiration and guiding hands, I would not have completed any of this. I'd like to thank CDR Matthew Humphries of the NPS Physics department for his assistance with sensor configuration and maintenance.

Many thanks go to my fellow master's students working on thesis research in the NPS Vision Lab: Maj. Brett Lindberg, LT Jason Nelson, LCDR Joshua Burkholder, and Capt. Justin Jones. The long hours spent in the lab and in the classroom were both frustrating and rewarding.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Modeling of an environment in three dimensions is used for a wide range of domains, including the mapping of terrain and structures from the air for civil and military planning, autonomous robot mapping and navigation, medical imaging and teleoperated systems, and augmented reality systems. The task specification for some of these applications requires a very detailed model to be developed in situ. Light Detection and Ranging (LIDAR) systems are used to acquire three dimensional (3D) surface data for modeling the environment. The advantage of LIDAR is the fast acquisition of a large amount of 3D point data with potentially high resolution. Current LIDAR systems typically rely on Global Positioning System (GPS) and/or Inertial Measurement Unit (IMU) data for external parameters to determine the pose of the sensor and to register scans. However, the data from GPS and IMU systems are generally much less precise than the LIDAR range data.

Ongoing research investigates methods to process the large amount of data produced by a LIDAR system in order to extract features usable to model the environment and detect objects for scene understanding. This research investigates methods of recognizing a fiducial marker within a LIDAR point cloud through geometric feature extraction. Detection and tracking of fiducial markers allows the LIDAR pose to be estimated using its own data and without relying on GPS/IMU data. A viable feature extraction method for fiducial detection in LIDAR range data provides a means for estimating the LIDAR pose to test ground vehicle navigation methods using simultaneous location and mapping (SLAM) techniques. Fiducials detected in both LIDAR range data and video image data also serve as control points for registering these two types of data.

A. ENVIRONMENT MODELING WITH RANGE INFORMATION

LIDAR range data measures the location of surfaces in the environment. The range data is normally formatted as range values or point locations in Cartesian or polar coordinates. A point cloud is a common term for the visualization of data in this format. Point representations alone are not always usable models of the environment for applications. The following sections describe the types of environments that are modeled with range information and various types of representations that are used.

Generating a 3D representation takes time and effort to collect all the needed measurements. For many mapping and modeling applications, the data is collected and processed separately, or “offline.” Dynamic changes to the environment are often not recorded and must be considered. Further difficulty comes with collecting accurate data in a hazardous environment such as a radioactive site, a disaster area, or a combat zone (Adams, 1998).

Several methods are available to model the real world with a computer. The type and structure of the environment being modeled affects the computational complexity and usability of a particular method. Environments with more man-made objects are sometimes easier to model with geometrically defined features. For example, the inside of a building frequently contains wall, floor, and ceiling surfaces that are generally orthogonal or at fixed angles to each other. The geometrically defined structural features in these environments provide a mathematical means of modeling the surfaces. A street along an urban corridor contains several man-made structure and ground surfaces mixed with natural ground and vegetation surfaces. This type of environment is characterized by a mix of man-made and natural surfaces. The unstructured environments in remote and unpopulated areas contain a majority of natural terrain and vegetation surfaces that are irregular (Burgard, 2008). An example point cloud image from a street environment is shown in Figure 1 with simple flat geometry of the road and very irregular geometry of the trees.

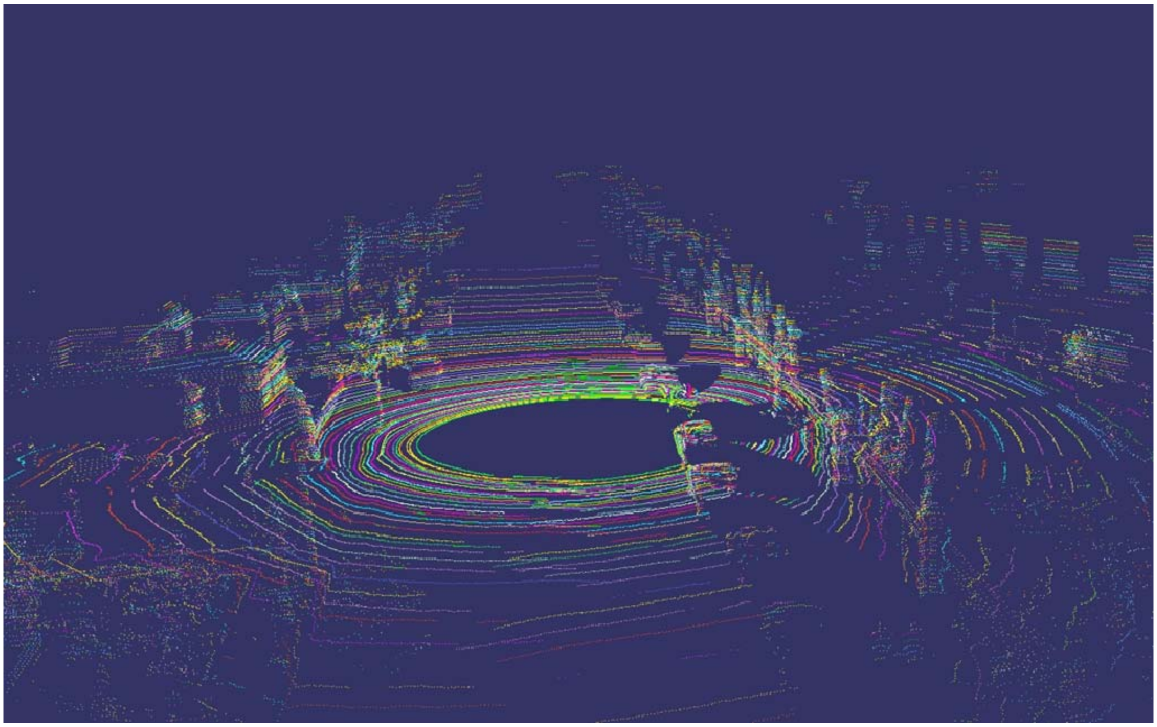


Figure 1. Point cloud representation of a street environment. The color variation indicates individual laser scan line patterns of the VLS. In this scene, there are on average 2200 individual range points per laser scan line with a total of 64 lasers.

B. LIDAR SYSTEMS

A 3D imaging system consists of a sensor to measure the distance to surfaces of objects in the environment and produces 3D coordinates or range and bearing values. The raw data can be easily represented by point clouds, with each point representing a spatial position and additional information such as color or laser intensity. Example 3D imaging systems include laser scanners, 3D optical scanners, 3D range cameras, LIDARs, and 3D flash LIDARs (ASTM Standard E2544-09b). More capable LIDAR systems produce high data point densities with accurate range information at a high sampling rate. Intensity values of the laser return are available from the receiver, but the resulting image may not be of high quality compared to common digital imaging systems.

Current applications of LIDAR systems include aerial mapping of terrain, measurement of structures, documentation and reverse engineering of public infrastructure, and robotic and autonomous systems for navigation and obstacle avoidance. An important topic of research for all of these applications involves the extraction of usable information from LIDAR point clouds (Wehr, 2005).

LIDAR systems vary in the type of laser transmission, the method to displace the laser through a scan pattern, and the type of environment the system is employed. Pulsed lasers measure the time of flight between transmitting the pulse and measuring the reflected return. Continuous wave lasers use either amplitude modulation or frequency modulation to measure the distance. A triangulation-based LIDAR uses a laser to project a laser spot on the object of interest and an offset camera that locates the laser spot and triangulates the range value. The laser is displaced using a dynamic mirror mechanism, a mechanism to move the entire laser, or through the motion of the platform to which the LIDAR is mounted.

During data acquisition, LIDAR systems are transported aboard air and land platforms. Aerial LIDARs generally scan a single laser beam in a whiskbroom pattern with a single axis scanning mirror. This results in a scan with a sweep width across the track of platform motion (Zhao, 2004). Airborne LIDAR systems are capable of mapping land and coastal areas in much shorter time than land based surveys. Federal agencies create airborne LIDAR surveys, such as the one shown in Figure 2, for mapping flood plains, earthquake zones, border zones, and forest canopies (Anderson, 2006). Terrestrial LIDARs use a fixed, sweep, or rotating scan pattern. The range is calculated either by a triangulation or pulse time of flight method. Example systems for terrestrial LIDARs include the SICK LMS-200, often researched for autonomous robot navigation, and the Velodyne HDL-64E S2 (VLS) employed in this research. Both systems are pictured in Figure 3.

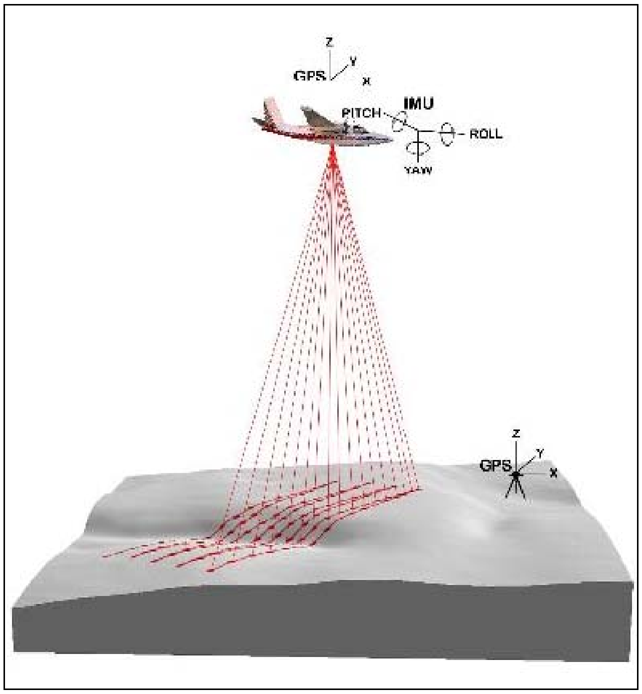


Figure 2. Aerial LIDAR system (ALS).

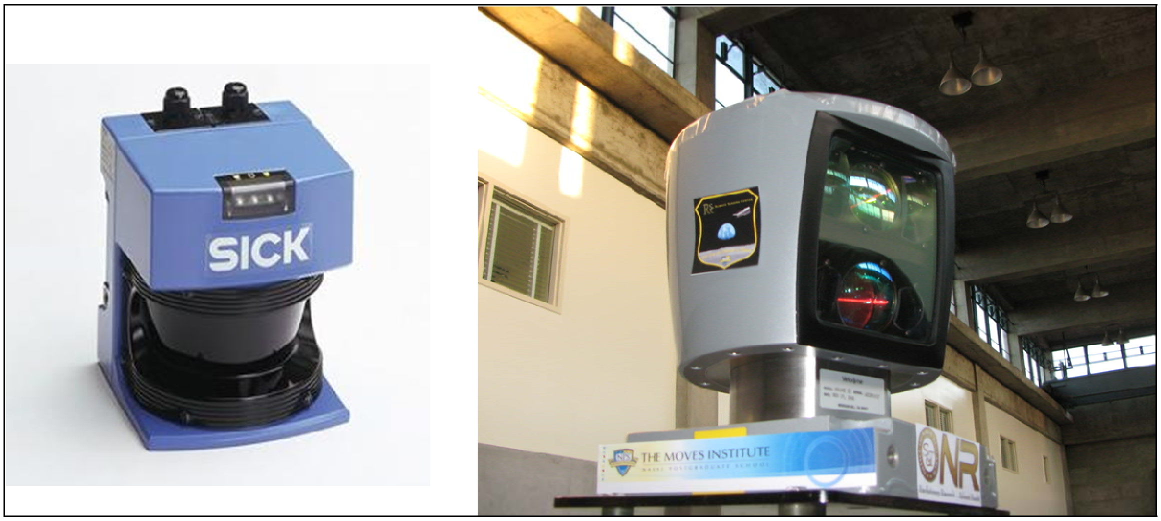


Figure 3. Terrestrial LIDAR System (TLS).
The SICK LMS-200 is depicted on the left and the Velodyne HDL-64E S2 is on the right.

C. 3D GEOMETRIC FEATURE EXTRACTION

LIDAR systems generate large sets of precise data. Without prior knowledge, the data contains only low-level information about the sampled environment: Only the 3D point values including location and reflectivity are known, but nothing about the underlying surface to permit making statements about the point relationships, particularly connectivity. Higher-level classification of the data is required to determine additional information such as surface definition, object detection, and terrain roughness. Extracting geometric features from the 3D data allows for a compressed representation that is more useful to other systems (Adams, 1998). Past work in fitting range data to geometric primitives shows that range data can be reduced by up to three orders of magnitude when represented as feature data (Faddema, 1997).

Extraction of geometric features, such as lines or arcs, involves fitting measured data to a model. How well a set of data fits the model justifies using the model as a representation of the data. There are two basic types of feature extraction. The first involves clustering the data in such a way that a good fit is determined, for example, with a Hough transform. The second utilizes least squares methods that attempt to minimize the differences between the observed data points and points that are located on the hypothesized feature. This is often used when fitting lines, circles, ellipses, and conic sections (Premebida, 2005).

Another distinction for feature extraction is between local and global methods. In general, local methods work bottom-up building the representation of the environment by starting with the raw data at a low level of contextual meaning. Global methods work top-down, being applied to the entire data set, and include an *a priori* understanding of the environment to extract the relevant features of interest from the data.

1. Segmentation

Because LIDAR data is a discrete set of range measurements, assuming the data represents a continuous surface often generates flawed geometric representations. Segmentation is a process to separate data points into groups based on a measure of similarity. The segmented data gives a more meaningful and easier to process representation. For example, a segmentation method may determine continuous regions of distance or intensity and detect the edges (discontinuities) between those regions (Premebida, 2005). Range data segmentation works better in geometrically structured environments, where there is a better fit to surface primitives, and not as well in unstructured environments due to the higher variance in surface structures (Besl, 1988, pp. 47 –61).

An example for global data segmentation is the split and merge technique. This method starts with the entire data set as an initial region of interest and continues to subdivide regions until each region satisfies some measure of similarity. The resultant data structure takes the form of a quadtree for 2D images, with each division resulting in four children, or an octree structure representing regions of 3D volumetric data with divisions of eight children.

D. FIDUCIAL MARKERS

Fiducial markers, also called control points, are reference markers that are recoverable by two independent sources of measurement. Fiducial markers can be used to register two sets of data and to determine the accuracy and precision of sensor output (ASTM Standard E2544-09b). The center of a circle and the center of a sphere are two examples for which the marker coordinates can be recovered based on the information in two separate data sets that include the marker surface. Examples of 2D fiducial markers include high contrast colored concentric circles, patterned matrix markers such as ARTag, and lines (Cawood, 2007). Three dimensional objects used for range image registration and pose estimation include cylinders, spheres, and orthogonal planes (Gao, 2007; Haas, 2005).

E. APPLICATIONS OF LIDAR 3D POINT CLOUDS

Robotics systems use LIDAR as a sensor for many applications ranging from object recognition on factory assembly lines, manipulation of pallets in warehouses, and as a mapping tool for navigation of autonomous robots. Simultaneous Localization and Mapping (SLAM) is an important robot navigation method that refers to the ability to determine an accurate localized position of the robot and an accurate map of the environment. Figure 4 shows an example of a robot generated navigational map. LIDAR systems are well suited for SLAM due to their accuracy. Several successful SLAM systems rely on artificial landmarks placed in the environment such as bar code reflectors, ultrasonic beacons, and visual patterns. This limits their usability in unknown environments. Other systems rely on external inputs from GPS and IMU sensors to estimate the robot's position and orientation.

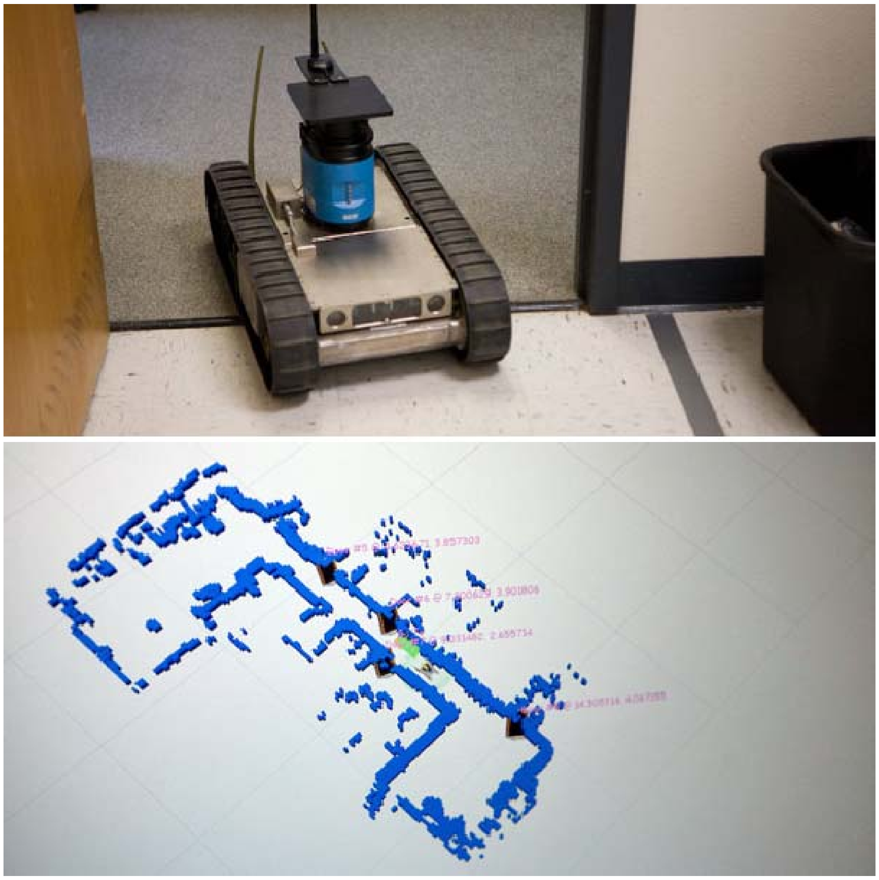


Figure 4. Robotic mapping using LIDAR
The 2D LIDAR on this autonomous robot was used to generate the map below for navigation in an indoor environment. (From Bullock, 2008).

Measuring and mapping infrastructure projects is another application for LIDAR systems. Both airborne and ground LIDAR systems have been employed to map city structures, survey coastal areas, and to assess highways (StreetMapper, 2010). Yoon (2009) describes a tunnel scanning LIDAR and associated feature extraction algorithms for an automated tunnel inspection process.

Future ground vehicle Augmented Reality (AR) systems offer the potential for increased situational awareness, access to geo-referenced mission information, and the ability for virtual area familiarization for relieving units. Augmented reality systems assist the user with a visual integration of information overlaid on the visual perception of the real environment (Lindberg, 2009). Registration is the alignment of the virtual augmentations (the virtual camera pose and internal parameters) with the user's real-world viewing parameters. AR systems have strict requirements to register the computer generated graphics with the true environment in position, orientation and camera lens characteristics (field of view, distortion, etc.). Registration errors result from poor tracking of the sensor and user viewpoints, an inaccurate calibration of the viewing device, or an inaccurate 3D environment model. Registration errors decrease the rendering quality and often the effectiveness of the AR information.

One drawback of many current real-time LIDAR mapping systems for military applications is the reliance on GPS/IMU as an external measurement basis for registration, localization, and navigation tasks. A viable natural landmark tracking method using LIDAR range data offers a potential means for registration of panoramic mapping of urban terrains, a source of pose information for autonomous and teleoperated robot navigation without GPS/IMU inputs, and the ability for registration of virtual content overlays with the view of the real world environment. Fiducial markers offer a way to determine the accuracy of ground truth feature extraction in LIDAR data for comparing methods that track natural landmarks. Landmark tracking methods with good performance can then register panoramic images with range data.

The thesis research presented herein investigates geometric fiducial marker shapes that can be recognized within LIDAR range data. A method of processing the LIDAR range data is implemented to extract these shapes and to determine their pose relative to the sensor. Experimental results are presented that investigate the effectiveness of the method to detect and estimate the pose of the fiducial markers at various ranges in an outdoor environment.

F. THESIS STRUCTURE

This first chapter introduces the pertinent topics of interest. Chapter II discusses the relevant background information and related work such as 3D fitting algorithms and geometric feature extraction from 3D point data. The research methodology is covered in Chapter III. Chapter IV explains the experimentation conducted to compare fiducial marker geometries and the overall accuracy and precision of the detection technique. The experimental results are presented in Chapter V, while Chapter VI offers a discussion of those results and future work. Finally, the conclusions of the research are presented in Chapter VII.

II. BACKGROUND

A. ENVIRONMENT MODELING

The method for modeling the real-world environment depends on the type of environment being modeled and the type of sensor data obtained from measuring that environment. Natural environments and man-made environments differ in the amount of geometric structure available for producing an accurate representation.

1. Structured Environments

Man-made objects and structures tend to have geometric features that can be described with fewer parameters, enabling geometric modeling methods to be more effective. In this thesis, an environment is called structured if it contains a majority of surfaces that can be represented by geometric features. Geometric features found in structured environments include the straight edges and lines of buildings, planar wall and object surfaces, and cylindrically shaped sign posts and electrical poles. Indoor artificial environments offer the ability to control lighting, the location of objects, and the amount and scale of dynamic changes to the environment. These controlled factors offer the ability to establish ground truth to develop and troubleshoot feature extraction methods. The interior of buildings, factory floors, medical operating rooms, and urban complexes are all examples of structured environments.

2. Unstructured Environments

In the case of an environment without geometric structures, other models and classification methods must be used. Natural terrain and objects such as rocks, trees, and bodies of water are characterized by uneven and porous surfaces that are not easily modeled with smooth geometric surfaces. For this thesis, these environments are called unstructured. The background composition and the locations of objects in naturally unstructured environments are not easily controlled making detecting and tracking objects in the environment more difficult (Burgard, 2008).

3. Mixed Structure Environments

A mixed structure environment is defined for this thesis as the combination of artificial man-made objects and naturally occurring objects. An example would be a moderate density populated area where there are man-made structures situated among natural terrain and vegetation. This research focuses on the mixed structure environment with the proposed fiducial marker geometries placed in outdoor environments that contain natural terrain and vegetation with artificial objects and structures.

4. Modeling Methods

A point cloud is a set of 3D points where each point represents the position of a surface. Surfaces are also modeled using a polygonal or a 3D geometric model. Polygonal meshes are collections of point vertices connected by edges to form a polygonal surface representation. Point clouds and polygonal meshes more often represent unstructured environments while geometric shapes such cubes, cylinders, and spheres are more frequently applied to structured and man-made environments. The modeling method affects the resolution, speed, and data size of the model.

B. LIDAR SYSTEMS

LIDAR systems use lasers to obtain range information. The lasers are either moved through a scan pattern by a mechanical means or reflected off of a moving mirror to scan a field of view (FOV). The relative range data from the sensor explicitly represents surfaces in the environment. Operation of LIDAR systems vary in the method of range data acquisition and the platform from which the system is operated.

1. Laser Range Data Characteristics

Range data from a LIDAR sensor gives an explicit representation of the surfaces in the environment within the system’s field of view. LIDAR systems measure the line of sight distance to surfaces using reflected laser energy and give only partial information about the nearest side of an object. The range and surface reflectance of objects determine the strength of the return signal detected. The 3D position is calculated using one dimensional range data and a two dimensional scan pattern. The range data set tends to be very large with false and missing returns causing incomplete data. There are a wide variety of LIDAR range systems with different range acquisition patterns. In general, it is more difficult to quickly generate LIDAR sampling patterns at rates and spatial densities comparable to those of devices producing 2D raster images. LIDAR actively emits the laser energy needed to measure the environment and interference effects must be accounted for resulting in a sparse or time lapsed image. Passive imaging systems only need to sense the energy in the environment providing a more simultaneous and dense image. The often relatively sparse nature of the LIDAR scan pattern increases the difficulty to segment and process 3D range data for features of interest compared to 2D raster images. The sparse sampling often does not provide data exactly on an edge that can be used to detect the edge. Despite a relatively sparse coverage of the field of view, the 3D range data enables segmentation methods not easily performed with passive 2D imagery.

a. Type of Operation

The method of laser operation for range data acquisition determines the parameters needed for data processing such as the resolution and sampling rate. The displacement of the laser through the field of view is accomplished in several ways. A single laser fired through a rotating mirror is provides a planar scan. Articulating the mirror along two axes produces a 2D scan pattern. Other system configurations mount multiple lasers inside a rotating unit. A whiskbroom scanner displaces the laser in an across-track scan pattern with a mirror in the direction perpendicular to the path of the sensor platform or vehicle. A push broom scanner, also called an along track scanner, uses a line of sensors oriented perpendicular to the track of the sensor platform.

b. Types of Lasers

There are a few methods with which the laser energy is measured to produce range estimates. A Time of Flight (TOF) pulsed LIDAR measures the elapsed time of flight between the transmission and detection of the reflection of a laser pulse. The speed of light and the measured TOF of the laser pulse permit range estimation. Pulsed LIDAR systems require high speed and precise components to transmit and measure the pulse. These components increase their cost (Adams, 1998, p. 27). Another type of LIDAR uses a frequency modulated continuous wave (FMCW) laser to calculate the difference between the frequency of the transmitted and received signals. This difference is directly proportional to range (Adams, 1998, pp. 42–43). Amplitude modulated continuous wave (AMCW) LIDAR measures the phase shift between the transmitted and reflected beams. This type of system is beneficial for close range applications (Adams, 1998, p. 27).

Flash LIDARs, or optical range cameras, are 3D imaging systems that detect a wide light flash with a focal plane array (FPA) detector to measure range. Both lasers and banks of Light Emitting Diodes (LED) can act as light sources. The FPA detects a flash illumination nearly simultaneously for all pixels in the frame. This type of system allows for high frame rate range data acquisition on the order of 30 frames per second, without any motion between the capture times for individual image points as is the case for scanning LIDARs (ASTM Standard E2544-09b).

c. Sources of Error

Range errors are determined by the laser pulse width, the performance of the laser detector, and the precision of the timing system that measures the time of flight for a reflected laser firing. Backscatter is the reflection of laser energy from gases and particles in the atmosphere that are not the intended surface target for ranging. If the backscatter energy is high enough, a false range measurement occurs. This, in turn, causes navigation errors in the estimate of the position and orientation of the sensor. These errors accumulate over time and affect the accuracy of the placement of the range readings taken within a sensor frame of reference into environment coordinates. For LIDAR systems dependent on pose estimation from other sensors such as an integrated GPS/IMU input, the pose estimation error affects all range point measurements. Further errors are introduced with the extrinsic calibration of the LIDAR. In the case of the VLS used for this research, calibration data is provided by the manufacturer. No further calibration attempts of the VLS were made. Finally, for the rotating mechanism of the VLS, a degree of error is introduced by the angular measurement of the rotation position at the time of laser firing. Theoretically, the potential exists for interference between multiple lasers with the same wavelength and pulse length when fired at the same time. Misinterpretation of the laser reflections results in erroneous time of flight and range measurements. This crosstalk between laser pulses is prevented in the VLS by both the sequential firing of individual lasers and by rotating the laser mount between firing lasers that are closely spaced.

Laser beam propagation is generally approximated with a Gaussian beam intensity profile, shown in Figure 5. This model describes the characteristics of the beam divergence, which increases with range, and the beam intensity, which decreases from the center of the cross section of the beam width. For applications that use a laser beam for range finding, the accuracy of the point of range measurement decreases as the laser beam width spreads (Alda, 2003). Further error can be introduced with effects such as overspill and underspill, where the laser beam is reflecting off of multiple surfaces simultaneously, yet only a single range reading is measured.

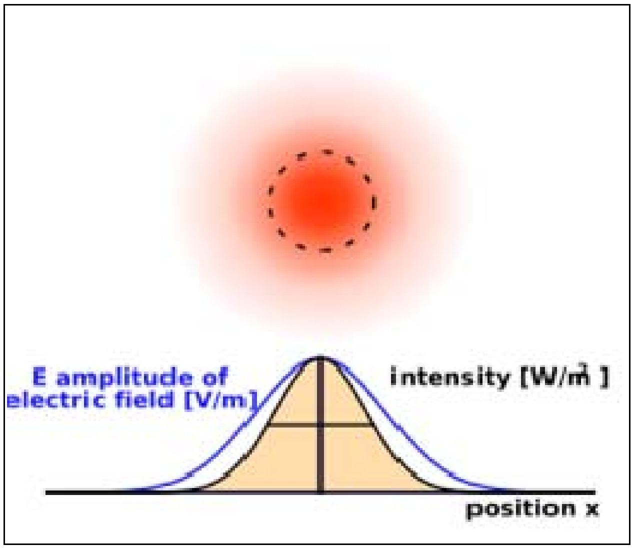


Figure 5. Gaussian laser pulse error model.
The intensity distribution of laser energy intensity is shown as a normally distributed value. (From Domenic, 2007)

d. Field of View (FOV) and Resolution

The Field of View (FOV) gives the angular coverage of the environment from the LIDAR origin measured in units of degrees in the horizontal and vertical directions. Resolution is the ability of a sensor to distinguish between two separate objects in a single scan. This separation is measured for the range, horizontal and vertical dimensions at a specified distance from the sensor. Angular resolution is used to remove the distance factor from the measurement (ASTM Standard E2544-09b).

The angular resolution is calculated as

$$angular\ resolution = \arctan\left(\frac{resolution\ dist.}{range}\right)$$

For a time-of-flight laser rangefinder, the pulse length determines the range resolution and accuracy. For the Velodyne HDL-64E S2 LIDAR, the range resolution is 1.5 cm (1-sigma) using a 905 nanometer wavelength laser with a 5 nanosecond pulse. The range resolution gives the distance along the range axis required to resolve two targets and is calculated below, where ΔR is the range resolution, c is the speed of light, T_p is the pulse length, σ_r is the range accuracy, and SNR is the signal to noise ratio of the sensor (Wehr, 2005).

$$\Delta R = \frac{c}{2}T_p$$

$$\sigma_R = \frac{c}{2}\frac{T_p}{\sqrt{SNR}}$$

2. LIDAR Data Processing

Compared to a raster image from a digital camera, it is generally more difficult to segment and process range data for features. A higher density of range points offers more data for feature detection, but generates a larger amount of data that must be communicated, processed, and stored.

a. Frame Definition

A frame is defined for this thesis as a region of interest for data acquisition. The size of the frame is specified as a range of the sensor rotation, up to the size of the sensor FOV limit of 360 degrees. In this thesis, a frame is normally all the data acquired within a full revolution of the sensor. The frame rate is the number of frames that are acquired per second, measured in frames per second (FPS) or cycles per second (Hz) (ASTM Standard E2544-09b).

b. Online Processing

Online methods aim to process data as it arrives, without intermediate bulk storage, in an effort to achieve real-time performance. The size and format of the data are critical to efficient online processing. Acquiring data takes time and delays results while waiting for the acquisition for the full frame size of data. Smaller sets of data can be processed as quickly as they are made available by the sensing system. Important to successful online processing is the need for the data to be presented in an order that allows efficient computation (Isenburg, 2008). Methods that process LIDAR data frames consisting of neighboring individual range values, 3D points, or network packets of point data offers the advantage of faster performance. The disadvantage is that at any one time, less complete data is available to extract useful information.

c. Post-Processing

Post-processing involves data manipulation at a time after sensor data acquisition. Data format and size are less critical than with online processing methods, since these steps can be modified as needed for the application in question. While post-processing methods may be more complete and accurate, the time difference between acquisition and obtaining results may preclude the use of these methods. As computational memory and algorithms improve, methods previously used for post-processing may become feasible for online processing.

C. GEOMETRIC FEATURE EXTRACTION FROM POINT DATA

Point data alone is of limited use. Higher-level abstractions of the environment features that are described by the data are more useful. Point data measurements are processed to determine, or extract, useable features. The located features can also be used as subsequent measurements for further detection of higher order features. Geometric feature descriptions are often used for 3D point data. Data sampled from man-made objects tend to fit geometric features better while natural terrain and objects such as trees are more difficult. Geometric features commonly fit to point data are line segments, circular arcs, elliptical arcs, simple polygons, boxes, cylinders, spheres, and cones.

Geometric features commonly used to model 3D environments include lines, circles, ellipses, spheres, cylinders, cones, and planes. These features can be defined either parametrically, shown in Table 1, or implicitly as in Table 2. Range data in the form of 3D points are used to fit an appropriate model of the geometric feature of interest. When a data segment fits a model of the feature well, the data is replaced with a parameterized instance of the feature. The environment, expressed in these features instead of raw data, is thereby compressed and often easier to process, render, and store.

Curve	**Equation**
Line	$ax + by + c = 0$
Circle, center(*a*,*b*) and radius *r*	$x^2 + y^2 - 2ax - 2by + a^2 + b^2 - r^2 = 0$
Ellipse (including circles)	$ax^2 + bxy + cy^2 + dx + ey + f = 0$ $where\ \ b^2 - 4ac < 0$
Hyperbola	$ax^2 + bxy + cy^2 + dx + ey + f = 0$ $where\ \ b^2 - 4ac > 0$
Parabola	$ax^2 + bxy + cy^2 + dx + ey + f = 0$ $where\ \ b^2 - 4ac = 0$
General conic section	$ax^2 + bxy + cy^2 + dx + ey + f = 0$

Table 1. Parametric definitions of geometric features.
(From Forsyth, 2002, p. 340)

Curve	**Parametric Form**	**Parameters**
Circle centered at the origin	$\left(r\sin(t)\ ,\ r\cos(t)\right)$	$\theta = r$ $t \in [0,2\pi)$
Circle	$(r\sin(t)+a\ ,\ r\cos(t)+b)$	$\theta = (r,a,b)$ $t \in [0,2\pi)$
Axis aligned ellipse	$(r_1\sin(t)+a\ ,\ r_2\cos(t)+b)$	$\theta = (r_1,r_2,a,b)$ $t \in [0,2\pi)$
Ellipse	$(\cos\phi(r_1\sin(t)+a)-\sin\phi(r_2\cos(t)+b),$ $\sin\phi(r_1\sin(t)+a)+\cos\phi(r_2\cos(t)+b))$	$\theta = (r_1,r_2,a,b,\phi)$ $t \in [0,2\pi)$
cubic segment	$(at^3+bt^2+ct+d, et^3+ft^2+gt+h)$	$\theta = (a,b,c,d,e,f,g,h)$ $t \in [0,1]$

Table 2. Implicit definition of geometric features (From Forsyth, 2002, p. 337).

1. Computer Vision Methods

An important problem that the domain of computer vision attempts to solve is the extraction of geometric primitive features from an image. Two common processes used for geometric feature extraction are the Random Sampling and Consensus (RANSAC) and the Hough transform methods (Bolles, 1981; Duda, 1972). Both methods can be applied to 2D and 3D data sets. Computer vision techniques also attempt to segment images into clusters of similar regions by using edge detection or object recognition algorithms in 2D visual images and 3D range images.

a. RANSAC

A technique commonly used in computer vision is called Random Sampling and Consensus (RANSAC) (Bolles, 1981). The RANSAC method attempts to filter out outliers—gross errors in the data set—prior to evaluating the fit of the data to a model. Least squares methods do not filter out outlier data and are susceptible to the level of accuracy of the data set to which they are applied. The essential steps of the RANSAC method are to:

- Make a hypothesis of the initial model parameters and eliminate data points that are outside a set error threshold.
- Using the remaining inlier data points, compute an improved model.

A model output from the method is considered to be the best fit model when weighted with the number of inlier data points used to fit the model and the measure of the resulting error of that model fit.

For RANSAC to be both robust and efficient, the data set must be as small as possible and still be able to determine the parameters of the model of interest. RANSAC methods work best with a high proportion of inliers, models with few parameters, and computationally efficient model fitting methods.

The background literature holds many examples of geometric primitive feature extraction from 3D data using the RANSAC paradigm. RANSAC methods are used by Tarsha-Kurdi (2007) to extract roof planes from airborne laser scanners. RANSAC fitting of cylinders to 3D data is shown in past work by Bolles and Fischler (Bolles, 1981), Chaperon (2001), and Bolles (1981). Schanbel (2007) proposed a more general method to detect planes, spheres, cylinders, cones, and tori in point-clouds.

One method of particular interest in detection of 3D geometric features from point data is to use domain knowledge to enhance the RANSAC algorithm. For example, past work in detecting roof planes from airborne LIDAR data uses domain knowledge of the mutual relationship of various roof plane combinations. The enhanced RANSAC method is able to improve detection of roof planes among other located planes that are not associated with building roofs (Tarsha-Kurdi, 2007).

b. Hough Transform

The Hough transform is used extensively in computer vision applications involving the detection of geometric features such as circles and lines from 2D images. Past work with Hough transform methods to detect objects in 3D point clouds includes Tarsha-Kurdi (2007) to extract roof planes from airborne laser scanners, Van der Glas (2002) for detecting the center and radius of spheres in 3D medical scanners, Rabbani (2005) to detect cylinders, and Ogundana (2007) for detecting spherical control points for calibrating multi-sensor shape measurement systems.

The Hough transform delineates a discrete parameter space that includes all possible parameter variations to accumulate potential features described by the available data. The frequency of an accumulated parameter combination gives the probable features located in the data set. This method requires enough correct data to work and a good choice for an accumulation grid because it is susceptible to noisy data. The parameter space can easily get large and as a consequence lose processing speed performance. The Hough transform is most frequently limited to application to 2D images for this reason, although it has been applied to detection of planes in 3D range images since the number of parameters is relatively small (Schanbel, 2007).

2. Mathematical 3D Feature Fitting

Several methods exist for fitting 3D points to geometric features that are similar to fitting methods for 2D images. The detection of circular and elliptical cross sections by fitting 3D laser scan line data is adversely affected due to partial and incomplete data available from the whole cross section. For the problem of fitting noisy data to a circle, there is a large variance in the fit as the arc subtended by the data on the circle approaches zero (Rusu, 2003).

a. Least Mean Squares Methods

Least Mean Squares methods, such as orthogonal distance regression, attempt to minimize the sum of squared distances from each measured data point to the nearest point on the hypothesized geometric feature (Atieg, 2004). A parameterized equation that describes discrete data can also be used, where the minimization of the error of the function parameters give the estimated fit.

Circular arcs are geometrically determined by at least three non-collinear points. If the points are located on a small arc segment, the resulting minimization matrix is poorly conditioned or becomes singular, preventing a (good) solution. For an algebraic fitting of conic sections, the implicit form of the conic can be used estimate the residuals that are to be minimized.

The equations to find a two dimensional circle fit include the parameters

x – the x-coordinate of the center of the circle.
y – the y-coordinate of the center of the circle.
r – the radius of the circle.

The distance function is defined as

$$d(x_i, y_i) = \sqrt{(x_i - x)^2 + (y_i - y)^2} - r$$

The objective function that is to be minimized to solve for the circle parameters is defined as

$$J(x, y, r) = \sum \left(\sqrt{(x_i - x)^2 + (y_i - y)^2} - r \right)^2$$

To find a circle fit to 3D points, a multi step process can be used (Shakarji, 1998).

- Compute the least-squares plane for the data.
- Rotate the data such that the least-squares plane is the x-y plane.
- Project the rotated data points into the x-y plane.
- Compute the 2D circle fit in the x-y plane.
- Rotate back to the original orientation.
- Perform a full 3D minimization search over all the parameters.

b. Taubin Method

The Taubin method for estimating planar curves in 2D or surfaces in 3D assumes that the generalized eigenvector provides a good initial estimate for iteratively minimizing the mean square distance of the data to the estimated feature (Taubin, 1991). Although this is a means of computing the estimate with a least squares approach, the iterative method is desirable for its ease of implementation in software. As with least squares methods, the accuracy is dependent on the amount of noise in the data and the arc length subtended by the data. Data that subtends a larger arc length return better fitting results (Rusu, 2003). An example MATLAB® implementation for this method is available that fits circles and ellipses to 2D data points (MATLAB® Central, 2009).

c Line Extraction Methods

Line extraction algorithms from point data include split and merge, iterative end point fit, line regression, RANSAC, and Hough Transform. Nguyen (2007) provides general descriptions of some common methods that as applied to indoor mobile robotics.

3. 3D Point Labeling and Classification

LIDAR-generated point data contains little meaningful information for most application purposes. Higher-level information about the points or groups of points is often needed. Past work in segmenting 3D LIDAR data into classes uses the descriptions of scatter points, linear points, and surface points. Lalonde (2006) developed a point cloud classification scheme for use in outdoor robotic navigation. The scatter classification represents porous volumes such as tree canopies and ground vegetation. The linear classification identifies long linear features such as wires and tree branches. The surface classification includes solid objects. Another 3D feature labeling method attempts to label each individual point or group of points as being a part of the ground surface. based on the absence of other data points in the vicinity under the point being labeled (Vandapel, 2003).

D. FIDUCIAL MARKERS

A fiducial marker is an object or marking in the environment that appears in sensor data that can be used as a control point from which subsequent measurements or comparisons of other environmental features in the sensor data can be made. Data captured from the same environment or scene can be correlated and registered using a common fiducial marker detected in both sets of data.

1. Applications of Fiducial Markers

Applications such as medical imaging or airborne geological surveillance combine data from multiple sensors using fiducial markers. Augmented reality and virtual reality applications employ fiducial markers either to correctly align virtual objects with the real environment or to measure the location of real objects to correctly position them in a virtual environment.

Examples of 2D fiducial markers include high contrast colored concentric circles, matrix pattern markers, and lines. Circular geodetic markers are used for verification of LIDAR systems by Hanna (2005) and 2D matrix markers are used by Cawood (2007) to overlay 3D computer graphics over live video images. The drawbacks of using 2D markers for LIDAR pose estimation include the ambiguity of co-planar surfaces in range data and the requirement to be placed in known locations prior to system operation.

Three dimensional objects used for range image registration and pose estimation include cylinders, spheres, and orthogonal planes. Steinbis (2008) positions 3D geometric cones on a flight line to register an outdoor augmented reality system and Zhao (1996) embeds spherical marker pins on a patient to register a teleoperated surgical device. Vertical cylinders are used for pose estimation of an autonomous wheelchair by Gao (2008). In this system, a LIDAR is used to detect the relative orientation of parallel cylinders on the wheelchair to dock the wheelchair for vehicle loading. Spherical markers provide ground truth data for a performance comparison of LIDAR sensors and for registering patient body movements during teleoperated medical procedures. These methods all require *a priori* 3D information about the environment.

2. Registration with Fiducial Markers

Registration is the process to determine the transformations needed to align two sets of data in a common global coordinate system (ASTM Standard E2544-09b). Registration aligns the virtual frame of reference with another frame of reference. Sources of registration error include system noise, environment model measurement, and temporal latency between data acquisition systems. In an unknown environment, model acquisition is a large source of registration error.

A local deviation is one that occurs at a particular region of registered data. A measurement of global registration errors is based on the normalized root mean squared error,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}}$$

The set of differences from the measured value and the reference value are aggregated with a representation of the amount of variance present using the same units as the individual measurements. Fiducial Registration Error is the error of fitting the data to a discrete target feature or data points from one frame to another frame (Cheok, 2005).

Registration of 3D point cloud data with image data from a 2D camera is sometimes completed by matching corresponding features in the two sets of data. The scene is often controlled, as in a factory or laboratory, to provide known features that are easily detected by both sensors to allow an accurate computation of the translation and rotation components of the registration transformation. Haas (2005) describes a wall-based registration method using three orthogonal planes to register the two data sets from one high and one low resolution LIDAR system. The plane parameters of two orthogonal walls and the ground are extracted to provide the registration transformation between the two sensors. Another transformation is computed to register the sum of the 3D LIDAR data with 2D image data from a video camera.

Registration techniques for augmented reality systems utilize 2D markers, 3D markers or no markers. Two dimensional fiducial markers include high contrast and matrix markers. Three dimensional fiducial markers include cylinders used for automated wheelchair docking and spherical tipped pins used for medical surgery. Cones have been used for registration of the video of a handheld AR device with objects on a flight line (Steinbis, 2008).

E. APPLICATIONS OF 3D POINT CLOUDS

Several domains contain applications of 3D point clouds. Reverse engineering makes use of 3D imaging to gain detailed measurements of objects and terrain. Computer graphics applications use point sets in the form of indexed vertex arrays of objects with surfaces constructed from meshes These meshes apply additional vertex property values to determine surface color, reflectivity, emissivity, and transparency.

Robotics applications of point clouds include sensing, manipulation, navigation, and mapping. Robot mounted LIDAR systems provide 2D planar or 3D point information about the environment that is an input to navigation and obstacle detection methods. This same data can also be applied to mapping functions. Lalonde (2008) describes past work involving robotic mounted LIDAR systems.

Augmented Reality (AR) systems rely on accurate registration of virtual information with the physical world to produce a form of mixed reality in which information from a virtual world is used to augment a user’s interaction with the real world. Registration techniques for augmented reality systems utilize 2D markers, 3D markers or no markers to derive relative pose estimates between the virtual and real environments.

There are two potential uses of LIDAR range data in an AR system: as a data source for registration and an online environmental modeling tool. Past work to register AR systems processes the video input to search for fiducial marker positions and make an estimate of the position and orientation of the video camera.(Steinbis, 2008). This method has the drawback that it requires prior knowledge of the 3D location of the fiducial points being tracked. The advantage of finding tracking features in LIDAR range data for registration is that the required 3D model of the local environment is also obtained by the LIDAR allowing operation in an unknown environment (Neumann, 2003).

F. RESEARCH FOCUS

This research investigates a method to determine the pose of a LIDAR sensor through the detection of fiducial geometric features in the range data. These features identify the location of fiducial markers placed in the environment. Types and sizes of fiducial markers are evaluated for effectiveness based on sensitivity and accuracy. Detection and pose estimation ideally are conducted in real-time data, but the experimentation in the current research is not constrained to the use of real-time data. Efficient techniques that use LIDAR range data for position and orientation determination of environmental features and real time environment map building would improve registration of LIDAR data with other imaging sensor data of the same environment.

III. METHODOLOGY

A. METHOD OF FIDUCIAL MARKER DETECTION

This chapter describes a method for three dimensional (3D) geometric feature extraction to detect fiducial marker location and orientation in LIDAR range data. Various 3D geometries are evaluated for suitability as fiducial markers that permit an estimate of the pose of the LIDAR sensor. The method attempts to detect geometric features within the individual laser scan line data. The sets of detected geometric features provide measurements for estimates of the location and pose of a specific 3D geometric fiducial marker. For cylindrical and cone fiducial marker shapes, the method compares sets of primitive arc features for fit to an *a priori* geometric model. Experimental test data captured with a Velodyne HDL-64E S2 LIDAR system are used to evaluate the accuracy of the method to detect cylinder and cone fiducial markers and the precision of the resulting fiducial pose estimates.

1. Design Overview

The approach for the design of the fiducial marker detector in this research follows a bottom up approach used by many pattern recognition systems (Duda, 2000, p. 10). This method follows a similar approach used to segment range data for mobile robotics (Premebida, 2005). Shown in Table 3, the processing steps start with the acquisition of 3D sample points of the relative position of the surfaces in the environment with a LIDAR sensor. The data output in the form of range data is pre-processed to remove any invalid or missing measurements. The segmentation step identifies and separates homogenous groups of points within intervals of range and gradient continuity for each laser scan line. Features extracted from the segmented sets of point data are used to provide a higher level description of each segment in order to classify the object surfaces that the point segments represent. The set of located feature properties is searched to locate clusters of features that fit the known fiducial marker geometry dimensions. The relative position and orientation of the fiducial marker from the LIDAR system are then estimated. If the fiducial marker pose in the environment is known, an estimate can then be made of the pose of the LIDAR within the same environment.

Processing Step	**Action, Measurement**	**Output, Features**
Sensor Output (3D Imaging System)	Range to environment surfaces	Range (Time Units)
UDP Network	Transfer blocks of laser firing data	UDP Packets
LidarInput (Software)		
Pre-Processing	Read UDP packets, convert range readings into Cartesian points	Cartesian Points (x, y, z, range, laser number)
Segmentation	Set point gradient properties and group sets of points from same laser scan line based on intervals of continuity	Segments (sets of points from same laser scan line)
Feature Extraction		
Arcs: circles	Fit segments to circular arcs.	CircularArc [center (x,y,z), radius, normal(x,y,z), RMS error]
Ground points	Estimate ground points with constant range in segment	Label points as constant range.
Wall points	Estimate points on objects that are not on the ground	Label points as constant gradient.
Axis lines	Fit arc centers to line, check fit for an axis line hypothesis	Line [point on line(x,y,z), axis direction(x,y,z)]
Feature Clustering		
Cylinders	Check cylinder fit of points in a set of arcs with centers that fit an axis line.	Cylinder [base center(x,y,z), radius, height, axis direction(x,y,z), RMS error]
Cones	Check cone fit of points in a set of arcs with centers that fit an axis line. Determine the apex angle with ratio of height to base radius.	Cone [base center(x,y,z), base radius, height, axis direction(x,y,z), RMS error]
Fiducial Detection	Based on fiducial marker definition, determine fiducial marker location and orientations.	Set of cylinder or cone features that match fiducial marker. Relative pose is location and orientation estimate of the fiducial features.
LIDAR Pose Estimate	Using known fiducial marker pose in environment, estimate LIDAR pose with difference from relative pose of detected fiducial marker.	

Table 3. Fiducial detection and pose estimation method.

2. 3D Imaging Sensor

The 3D imaging sensor used for this thesis is a pulsed LIDAR design (HDL-64E S2 User's Manual) which employs a head that rotates around the vertical axis with 64 lasers mounted at fixed vertical offset angles. The resulting scan pattern swept out by each laser is a cone that intersects surfaces in the environment. As a consequence, much of the laser scan line patterns across flat surfaces are not straight lines, but rather arcs, or conics. Line segment fitting is a common method for detecting structured objects in the planar scan pattern of 2D LIDAR systems, for example, on indoor autonomous robotic systems. However, due to the conical nature of the VLS laser scan line pattern, there are fewer straight line segments in the data, even along flat and linear object surfaces. Scan lines that intersect a long flat wall tend to have a curved shape that matches the curve of intersection between the wall plane and the cone of the laser scan.

3. Sensor Network Communication

Output from the sensor is via a User Datagram Protocol (UDP) network accessible with a category-5 network cable. The UDP network protocol sends datagram messages with no guarantee of packet delivery or packet transmission sequence. Packets of laser range data are formatted inside the sensor and sent over the network in the order that they are created. Packets are processed at the receiving end in the sequence that they arrive, yet older, out-of-order packets are dropped. The advantages of using a UDP network communication protocol are the speed of communication and the ability to multicast packets simultaneously over the network to several receivers.

4. Sensor Data Processing Input

Data output from the VLS is formatted such that range data is packed in the sequence of acquisition and not according to any sort of spatial arrangement. This means that although two data points are acquired at nearly the same time, differences in laser angle and sensor rotation at the time of firing preclude a simple spatial association between the data. Over a series of sequential data packets, neighbor range points from the same laser can easily be accessed. Identifying which range points are vertically near each other is more difficult as they stem from different lasers whose beams cross. For this reason, much of the range data processing operates on sequential points within the individual laser scan lines. These laser scan lines are from a single laser at fixed vertical and horizontal offset angles relative to the VLS rotation angle. The VLS system rotates clockwise (CW) with rotation rates variable from 300 revolutions per minute (RPM) up to 900 RPM. This is equivalent to a range of five cycles per second (5 Hz) up to 15 Hz.

a. Error Model

For this research, the assumed error model for the laser range data follows a Gaussian distribution with a zero mean and a variance of σ_r^2. This is a commonly used error model in laser range finding devices (Wehr, 2005). The VLS manufacturer performs a calibration procedure to derive correction factors for each laser that enable a more accurate interpretation of the 3D location of each range return position. This calibration is used to transform the raw range value to a 3D Cartesian point.

b. Wall Order

The positions of the 64 lasers in the device do not correspond monotonically to the vertical laser direction; i.e., a lower-positioned laser can be directed with a fixed angle pointing higher up than a laser mounted above it on the sensor. Hence, laser beams can cross over vertically. The *wall order* of the laser firings as defined for this research is the sequence of laser hit points as they would occur on a vertical wall at least five meters from the laser. For this thesis, range values less than 5 meters are not considered. Wall order is defined as a numbering sequence from top to bottom [0,63] and is solely determined by the vertical angle. This ordering makes more intuitive sense for determining the relative position of neighboring lasers and their resulting data points. The VLS manufacturer defines the variable *vertCorrection* as the vertical angular deflection from the horizon, measured in degrees, for each laser. A positive value indicates an offset above the horizon and a negative value below the horizon. The wall order is calculated by ordering the calibration file *vertCorrection* values from maximum to minimum.

5. Processing Software

Three software projects provide the ability to generate test point clouds, read and record VLS input online, and read and process VLS data offline. More work is required to fully integrate the processing methods into the online system.

The OpenSceneGraph (OSG) 3D graphics toolkit is used for rendering and visualization of the processing input and output. The OSG toolkit is written with C++ on top of the OpenGL 3D graphics application programming interface (API). The graphics processing nature of OSG allows for direct transformation of geometric primitive features detected in the LIDAR point data to a rendered graphical visualization.

The VLS data processing flow is shown in Figure 6. Input sources from either a live LIDAR network connection or a recorded packet capture file can be processed with this software. Processing and output are controlled by various threshold and mode option settings, The VLS calibration file is read in to memory to provide correction values for all of the lasers.

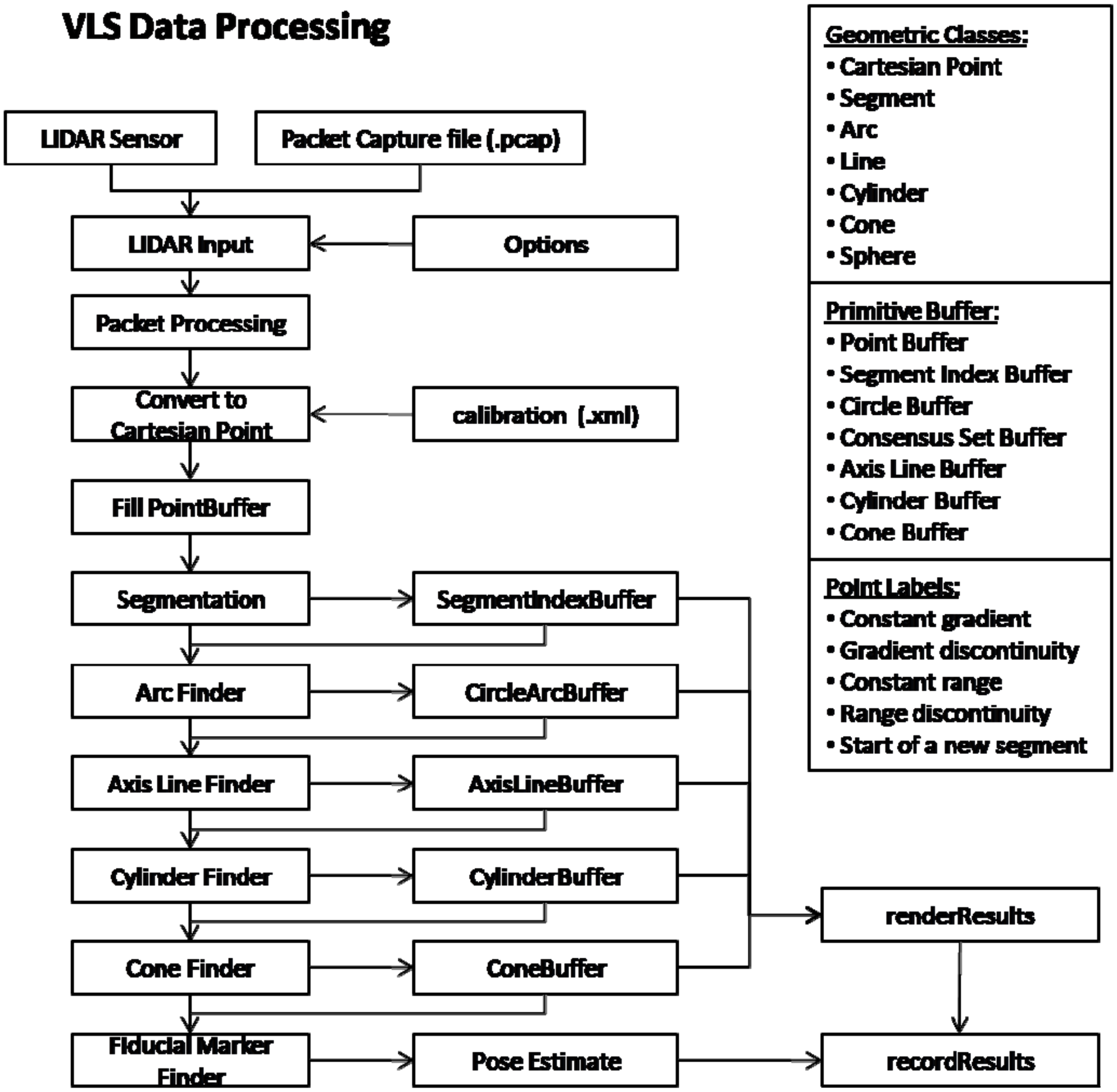


Figure 6. VLS data processing flowchart.
This flow chart depicts the full system starting with LIDAR data from the sensor or a recorded packet capture.

a. Data Structures

The VLS data packet contains the range and intensity values for six sequential firings of each laser. The rotational position of the sensor is tagged with the data packet so that the 3D position of each range value can be determined. As data packets are read by the processing software, a point buffer is filled that contains point objects that include the Cartesian coordinates, range, intensity, and flags for gradient type and segment position.

The concepts of point, firing, packet, and frame organize the data in an intuitive way: Each *frame* contains all of the points from each laser for one rotation of the VLS. Every processing *packet* contains the same information as one UDP network packet. A *firing* corresponds to the sequence of firings from one block of 32 lasers such that there are 32 individual data points. There are two blocks of 32 lasers for a total of 64 lasers in the system. A *point* is the return information from a single firing of a laser.

The VLS data is read live from the sensor, from a recording of captured packets, or from a logfile. The live input is via a UDP network. Packet captures are recorded from the network in the same format as UDP packets. The log file contains the same point, firing, packet, and frame organization as the UDP packets, but is formatted as a delimited text file. Log files of representative data can also be generated with the Python version of the *LidarSimulation* and a 3D graphical scene in the Delta3D game engine.

The *PrimitiveBuffer* class contains several buffers to temporarily store data from the input and intermediate processing steps. Member data structures of the *PrimitiveBuffer* class include the *PointBuffer*, *SegmentIndexBuffer*, *CircleBuffer*, *ConsensusSetIndexBuffer*, *AxisLineBuffer*, *CylinderBuffer*, and *ConeBuffer*. The point buffer stores the input of points for an entire frame of data input. After the segmentation step, the segment index buffer stores the point buffer indices of the start of each segmented set of scan line points. The segments are checked for fit to a circular arc and the results are contained in the circle buffer. The consensus set index buffer stores sets of circular arcs with centers that fit a hypothesized axis line. Axis line hypotheses with a good enough fit are stored in the *AxisLineBuffer*. Finally, the set of data points that correspond to arc center points that define the axis lines in the buffer are tested for fit with a cylinder or cone. The cylinder or cone hypotheses with good enough fit to the data are added to their respective buffer. Rendering methods are able to access any of the buffers to graphically display the features or other visualizations. Pose estimates are determined from the cylinder or cone objects found in the environment.

b. Geometric Class Descriptions

Several geometric primitive classes are defined to contain the dimensions and descriptors of features used throughout the detection and estimation process. Table 4 describes the primitive classes implemented for this thesis.

Class	**Members**
Cartesian Point	Coordinates $[x, y, z]$, horizontal gradient, continuity type, valid flag, Laser number
Segment	list of Cartesian Points
Arc	Center $[x_c, y_c, z_c]$, Radius RMS error
Line	Center $[x_c, y_c, z_c]$, Normalized Axis Direction $[x_a, y_a, z_a]$ Segment of points on the line
Cylinder	Center $[x_c, y_c, z_c]$, Normalized Axis Direction $[x_a, y_a, z_a]$ Base Radius Height RMS error
Cone	Center $[x_c, y_c, z_c]$, Normalized Axis Direction $[x_a, y_a, z_a]$ Base Radius Height RMS error
Sphere	Center $[x_c, y_c, z_c]$, Radius RMS error

Table 4. Geometric class definitions.
The listed geometric features and their associated data members are used for the segmentation and feature extraction processes.

The **point** class defines a Cartesian point in 3D. Values for the laser number, range, intensity, gradient type, and segment position are included. Points are a fundamental primitive used throughout the processing chain. For this reason they are stored in a buffer upon reading from the data input source and accessed through pointers.

The **segment** class defines a horizontal sequence of points fired from the same laser that have a homogenous range or gradient. Each segment is a set of Cartesian points that is defined by an index to the first point in the segment along with the number of points in the segment. This method of indexing reduces the size of the buffer needed for segment definition.

The **arc** class describes a circular arc. Each segment is tested for fit to an arc within the thresholds set for the processing mode. The arc class contains the center point of the arc, the radius, an estimate of the error of fit, and a reference to the segment of points that were used to fit the arc. Arc objects are stored in an arc buffer used by subsequent fitting functions.

The **line** class is used to define axis lines for fiducial marker hypotheses. Each line object contains a normalized direction vector stored as a Cartesian triple value and a Cartesian point on the line. Axis lines hypotheses are fit using the center points of clusters of arc segments and stored in the axis buffer when the fit is determined to be good enough.

The **cone** and **cylinder** classes are used to describe geometric features in the environment. These shapes correspond to the fiducial marker geometries tested. The cone and cylinder classes include a center point, axis line direction from the center point, base radius, height, and aspect ratio. The aspect ratio is defined as the height divided by the base radius. This aspect ratio is proportional to the half-angle of the apex of the cone and permits comparison of cone apex angles. Objects of each geometric shape are stored in a corresponding buffer to enable referencing for rendering or fiducial marker search functions.

6. Segmentation

Segmentation of each scan-line of point data according to a threshold for range discontinuity results in a set of *Segments*. Each *Segment* represents an interval of continuity and contains points from the same laser scan line in sequential order of firing. The end points of the *Segment* are against a range discontinuity above a threshold.

Invalid points are defined as points that do not represent a valid range sample. A pulsed LIDAR system includes a minimum time of flight threshold to avoid backscatter from environmental obscurants and maximum time interval based on the maximum effective range to detect a reflected laser pulse. An invalid point is marked with zero distance, occurring when the laser detector does not receive a reflected laser pulse within the time interval required for a maximum range return or receives energy in a time interval less than the minimum range threshold.

Here, the range data is segmented with 1D range discontinuity filters. The first layer of segmentation occurs at a low level, that is, on the raw range measurements from the LIDAR. Initial hypotheses for classification of points into intervals of continuity, intervals of constant range, and range discontinuities (large gradient) are made based on a series of 1D filters, shown in Table 5. Points are flagged with the segmentation results to be used by higher processing levels.

<table>
<tr><th>Filter Design</th><th>Function</th></tr>
<tr><td>$\begin{array}{|c|c|c|}\hline 1 & 1 & -2 \\ \hline r_{i-1} & r_i & r_{i+1} \\ \hline\end{array}$</td><td>Continuity type:
• gradient = 0 → constant range (ground)
• gradient < 0 || gradient > 0 → constant gradient (walls)</td></tr>
<tr><td>$\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline r_{i-2} & r_{i-1} & r_i \\ \hline\end{array}$</td><td>segment index buffer values:
• set new segment flag
• fill start new segment index values in buffer</td></tr>
<tr><td>$\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline r_{i-1} & r_i & r_{i+1} \\ \hline\end{array}$</td><td>Horizontal gradient flag values:
• gradient = 0: zero gradient flag
• gradient > 0: positive gradient flag
• gradient < 0: negative gradient flag</td></tr>
</table>

Table 5. Segmentation filters.
The gradient at r_i is calculated from its adjacent points or from the two preceding points with the weighting shown in the top row of the filter design depiction.

The effectiveness of 1D filters for segmentation depends on the choice of threshold values. Larger threshold values discriminate less between surfaces that have similar range gradients, while smaller thresholds are affected by noise and rough object surfaces. Segmentation is improved with a threshold level that adapts to the expected differences in noise, range, and surface complexity. The segmentation method for each laser scan line is described in Table 6. Series of sequentially fired points for each laser are grouped together in containers called *Segments* if there is a low enough gradient difference between them. Where there is a large enough range discontinuity, the gradient calculation will identify the corresponding points as edges. The hypothesis is that the Segment containing points that exhibit a low gradient difference among them represents data points from a common surface. This Segment is later tested in subsequent steps.

Algorithm: Scan Line Segmentation

```
begin, initialize: segment, segmentIndexBuffer
    segment ← is not open
    store last two points in the scan line sequence
for all points in the laser scan line:
    gradient ← apply 1D filter to the points
    point ← set gradient flag based on gradient value
    if ( segment.notOpen ) :
        segment ← add this point
        segment ← open the segment
    else if ( gradient == 0 && segment.isOpen ) :
        segment ← add this point
    else if (gradient != 0 && segment.isOpen ) :
        segment ←add this point
        segment ← close the segment.
        segmentIndexBuffer ← store segment start index
end
```

Table 6. Laser scan line segmentation algorithm.
All segments start with a closed flag which is changed to open when the first point is added. A segment is flagged as being closed if it is currently flagged open and the computed filter gradient is larger than the threshold setting.

Segments of points belonging to the same interval of continuity are passed on to feature extraction steps. Points belonging to intervals of constant range are hypothesized to be a member of a ground surface and therefore not a member of an object that is potentially the fiducial marker of interest. These sets of points are culled from those passed to feature extraction steps to improve efficiency. The thresholds used to determine segments of constant range and intervals of continuity determine the number and size of segments output from the segmentation step.

7. Feature Extraction

The feature extraction step uses the sets of points classified as segments to produce hypotheses for extracting geometric features such as arcs. In the case of the rotating VLS sensor, each segment of points originates from the scan line of one laser as it rotates through a scene. The scan line intersection with spheres, cylinders, and cones has the shape of an arc segment. The fiducial detection method starts with estimation of arc segments.

The detection algorithm uses the domain knowledge of the scan line intersection parameters to find features of interest in the point data. The first level of feature extraction is to segment the scan line points into intervals of continuity, where there is a near constant gradient between segment point ranges, or intervals of constant range, where there is a near constant range between segment points. The arc extraction method operates on these segments. Further feature extraction steps cluster the arc features to extract cylinder and cone features by using the parameters of the shape of interest and the mutual relationships of neighboring arc features.

a. 3D Circular Arc Fitting Algorithm

The following method is implemented to fit a series of 3D point data to a circular arc. This method is similar to the process described in Shakarji (1998). For short arc lengths of laser scan line data, the assumption is that the data lie approximately on the same plane. A least-squares method fits the 2D point data to a circle, and the result is transformed back to the original 3D coordinates. The circle fitting method is an iterative method that is based on a least-squares solution that does not require the use of matrix mathematics and avoids complex computations (Taubin, 1991). An example MATLAB® implementation for this method is available that fits circles and ellipses.

8. Feature Clustering and Classification

The resultant arc features from circle fitting provide the measurements for geometric shape detection. Clustered arc features define a hypothesis for the surface of a geometrically shaped object, represented by a set of points. For the detection of a known fiducial marker, the number of geometric hypotheses can be constrained with the marker parameters and domain knowledge of the environment. In this thesis, for the case of data that is acquired with a rotating laser, two points or features are close enough to be considered for a hypothesized feature if their center points are within 10 degrees of LIDAR rotation from each other. This limits the number of points and features that are tested for fit to a 3D geometric feature.

A RANSAC approach searches for model fit of feature clusters to the fiducial marker model of interest. The sets of neighbor arc features described above are then tested for fit to a cylinder or a cone. The initial step of determines inliers by comparing the radius of the arc feature and the orthogonal distance from the center of the arc to the hypothesis axis line. Figure 7 shows the distance, d , from a 3D point being tested, X_o , to the hypothesis line with endpoints X_1, X_2 . This distance is calculated as

$$d = \frac{\left\| (X_0 - X_1) \times (X_0 - X_2) \right\|}{\left\| X_2 - X_1 \right\|}$$

The symbol × denotes the vector cross product and the symbol $\| \ \|$ denotes the Euclidean norm of a vector.

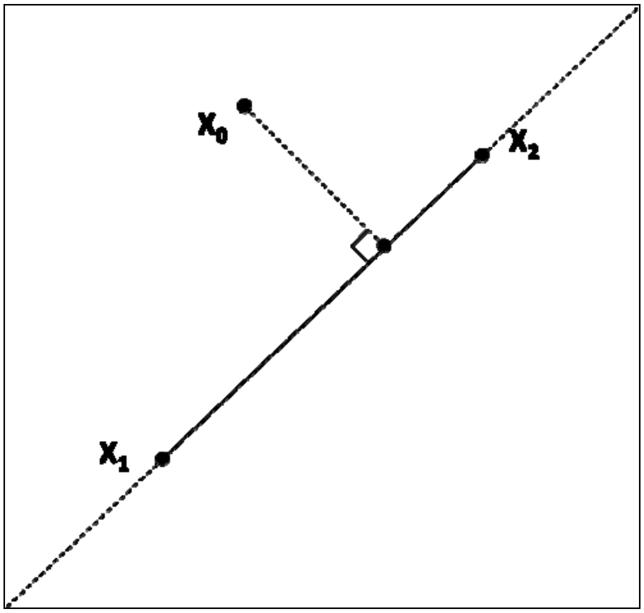


Figure 7. Orthogonal distance calculation from a 3D point to a line.

Once the set of inliers is determined, the axis line is redefined with the consensus set of arc center points and performing an orthogonal distance regression. To compute this, first find the centroid of the point data and then form a matrix A, where each row are the point coordinates x, y, z shifted by the mean centroid value. For the line of best fit to a set of 3D points, solve for the eigenvector of matrix A that corresponds to its largest eigenvalue using an SVD decomposition of A where

$$\mathrm{A=UWV}^{t}$$

such that U is an orthogonal matrix, the diagonal values in W are the singular values, and V^{t} is the transpose of the matrix containing the eigenvectors of matrix A which are also the principal components of the point data. The 3D line of best fit includes the centroid of the data and the direction vector is the eigenvector described above. For further discussion of the method of 3D line fitting with singular value decomposition see (Golub, 1996, pp. 70-73).

With an axis fit from the arc centers in the consensus set, the fiducial marker geometry is estimated using the arc feature radii. This geometric model is then tested for goodness of fit by calculating the resulting error of all the points that constitute the axis line arc features. Acceptance of the geometric model is gauged using this error estimate and thresholds set to minimize spurious geometric models that fit well to noise.

To improve the RANSAC method performance for fiducial marker detection, knowledge of the fiducial marker size and shape and the laser scan pattern of the VLS limit the number of random hypotheses. Table 7 describes the individual thresholds and Table 8 shows the entire algorithm.

Threshold Definitions

$N_p =$ initial set of points for an axis hypothesis

$N_{arcs} =$ minimum number of arcs required for cone or cylinder hypothesis

$K_{\max} =$ maximum number of iterations

$e_{shape} =$ error tolerance for point fit to geometric shape hypothesis

$e_{circle} =$ error tolerance for points fit to a circular arc

$h_{cone} =$ max cone height threshold to eliminate tall, thin cones
that fit cylinder data points

$r_{\max} =$ max radius to contain search for fiducial marker

$r_{\min} =$ min radius to contain search for fiducial marker

$t_{rot} =$ rotation window size for clustering circle arcs

$t_{line} =$ maximum inlier distance for arc center point to an axis line

$t_{orientation} =$ cosine of difference angle between orientation directions

$t_{aspect} =$ max difference in aspect ratio, the height/base_radius

$t_{horizontal\_gradient} =$ horizontal gradient threshold for constant range flag

$t_{segment} =$ gradient threshold for segmentation flag

Table 7. Threshold definitions.
These thresholds set the fiducial detection and pose estimation method constraints.

Algorithm: (RANSAC approach for 3D fiducial marker detection)

```
begin initialize:
for all arc feature center points:
    determine set of neighbor arcs using $t_{rot}$
for all sets of neighbor arc clusters of size > $N_{arcs}$ :
    while k < $K_{\max}$ do:
        choose 2 arc centers
        line ← calculate a hypothesis axis line
        for all remaining points in the neighbor arc set:
            form a consensus set of circle arcs for an axis line
            if center point distance to axis line < $t_{line}$
                then include arc in the consensus set
        if consensus set size > $N_{arcs}$ :
            axis ← refit axis line from consensus set arc center points
            for all arcs in consensus set:
                rms_error ← 0
                for all points in the arc:
                    rms_error ← rms_error + point fit error to fiducial model
            if rms_error < best fit consensus set
                then this set is the new best fit set
            remove consensus set arcs from the set of features for this iteration
        next iteration, k = k + 1
return best fit consensus arc feature set
if fiducial hypothesis base_radius < $r_{\max}$ and base_radius > $r_{\min}$ :
    $\vec{v}_1$ ← fiducial hypothesis orientation
    $\vec{v}_2$ ← fiducial known orientation
    if $\vec{v}_1 \bullet \vec{v}_2 < t_{orientation}$ :
        $h$ ← height of fiducial hypothesis
        $r$ ← radius of fiducial hypothesis
        if cylinder fiducial hypothesis:
            then valid cylinder fiducial
        else if cone fiducial hypothesis:
            if $\frac{h}{r} - f_{aspect} < t_{aspect}$ and $h < h_{cone}$ :
                then fiducial hypothesis is valid cone fiducial marker
end
```

Table 8. RANSAC approach to fiducial marker detection.

A region growing method was ruled out due to the reliance on implicit connectivity in the data. The data generated from the VLS LIDAR has a high horizontal density and a sparse vertical density. With only 64 laser scan lines, the vertical separation between scan lines increases the potential for error if vertical connectivity associations are assumed incorrectly.

9. Cylinder Similarity Measurement

A cylinder similarity measurement is used to identify clusters of cylinder hypotheses that describe the same cylinder shape. Each cluster is combined into a single cylinder hypothesis.

Two cylinder hypotheses are considered similar and likely to be from the same surface in the environment if the following conditions are true:

- Radius: the radii of the two cylinders are similar

$$\left| r_{fiducial} - r_{estimate} \right| < t_{similarCylinderRadius}$$

- Axis orientation: the axis orientations of the cylinders are near parallel. If $\vec{v}_1$ and $\vec{v}_2$ are the two axis vectors and θ is the angle between them, the dot product is defined as

$$\vec{v}_1 \bullet \vec{v}_2 = \left\| \vec{v}_1 \right\| \left\| \vec{v}_2 \right\| \cos\theta$$

- For parallel vector directions, the dot product equals one. For orthogonal vector directions, the dot product equals zero. Two cylinder axes are determined to be similar if the dot product of their axis vectors is close to being equal to 1.0, where

$$1 - \left| \frac{\vec{v}_1 \bullet \vec{v}_2}{\left\| \vec{v}_1 \right\| \left\| \vec{v}_2 \right\|} \right| < t_{axisOrientation}$$

- The point distance from the center of each cylinder to the other cylinder’s axis line is close enough. For a 3D line defined by two points

$\mathrm{x}_1 = (x_1, y_1, z_1)$ and $\mathrm{x}_2 = (x_2, y_2, z_2)$, the orthogonal distance from any point $\mathrm{x}_0 = (x_0, y_0, z_0)$ to the line is

$$d = \frac{\left|(\mathrm{x}_2 - \mathrm{x}_1) \times (\mathrm{x}_1 - \mathrm{x}_0)\right|}{\left|\mathrm{x}_2 - \mathrm{x}_1\right|}$$

- Two hypothesized cylinder axis lines are close enough to be estimates of the same cylindrical shape in the environment if

$$d < t_{centerDist}$$

10. Cone Similarity Measurement

Cone hypotheses are compared for similarity in much the same way as cylinders with the addition of a comparison of the angle of the slope of the sides of the cone. Cone shapes are considered similar if the following are true:

- Axis orientation is close enough, calculated the same as for cylinder axes.
- The point distance from the center of each cone to the other cone's axis line is close enough.
- The ratio of the radius of the base to the height of the cone is close enough to the same ratio for the fiducial marker of interest. This ratio is equal to the inverse tangent of the apex angle φ, at the top of the cone, where

$$\frac{r_{base}}{h_{cone}} = \arctan\left(\frac{\varphi}{2}\right)$$

11. Fiducial Marker Detection

A detection of a cylinder or cone fiducial marker is determined by the best fitting hypothesis among the set of cylinders or cones found. For cylinders, the set of cylinder fiducial hypotheses is searched for the best fit to the fiducial marker parameters. A distance is calculated based on the absolute difference between the hypothesized root mean squared error of fit to the data points and the difference in the radius estimate. The cylinder hypothesis with the lowest distance is returned as the system-determined fiducial marker location and orientation relative to the sensor.

For this research, the ratio of the height to base radius is called the aspect ratio. The set of cone fiducial hypotheses is searched for lowest distance cone hypothesis based on the root mean squared error of fit to the data and the difference between the fiducial aspect ratio and that of the hypothesis cone. The best fit cone hypothesis is returned as the system-determined fiducial marker location and orientation relative to the sensor.

12. Processing Method Output

The objective of this research is to process 3D point data to recognize a known fiducial marker in the environment. The output of the method is the location and orientation of detected fiducial markers and the parameters of the model that best fit the data. Since the method extracts several geometric features in the search for the fiducial marker, these features are also available for modeling the environment.

a. Fiducial Marker Pose

The fiducial marker pose is given by the location of the center point and the direction of the central axis. For a cylinder fiducial marker, the center point is the Cartesian coordinate location, $\left(x_c, y_c, z_c\right)$ for half the distance from the base to the top along the cylinder axis. The orientation is given as a normalized direction vector, $\hat{n}_i$, with normalized Cartesian components $\left(x_i, y_i, z_i\right)$.

B. SIMULATION OF LIDAR DATA ACQUISITION

Synthetic environments have been used in past research into geometric shape extraction from 3D point cloud data. Synthetic point data sets were used to test virtual representations of real-world disaster scenes by Biggers (2009) and Bae (2004) tests automated point cloud registration techniques. The advantage to using simulated point clouds for testing proposed processing methods are that the point cloud data set size can be constrained and derived from simulated environments of know dimensions. The addition of noise to the synthetic data set further represents the real world data.

The generated data is not intended to replace experimental data. The simulation provides a sandbox in which to verify and experiment with various point cloud feature extraction algorithms. The data is representative of VLS generated data. Inside the simulation it is possible to control a set of geometric features with known parameters, location, and orientation. The ability to control test scenes of varying complexity for variations in object shape, location, and orientation permits the verification of the coding of processing algorithms (Baltzakis, 2003).

1. Simulation Implementation

The 3D LIDAR simulation created for this research provides a means to verify code implementation and experimental setups in a controllable manner. The process displayed in Figure 8 implements a simulated VLS system using Python bindings for the Delta3D game engine. The simulation produces point cloud data sets according to the actual VLS laser firing parameters to simulate laser hit points on the surfaces of graphically rendered objects within the Delta3D scene. Depth testing in the virtual world replaces the VLS laser firing by calculating the range value for a line of sight intersection along the laser firing direction with the scene geometry.

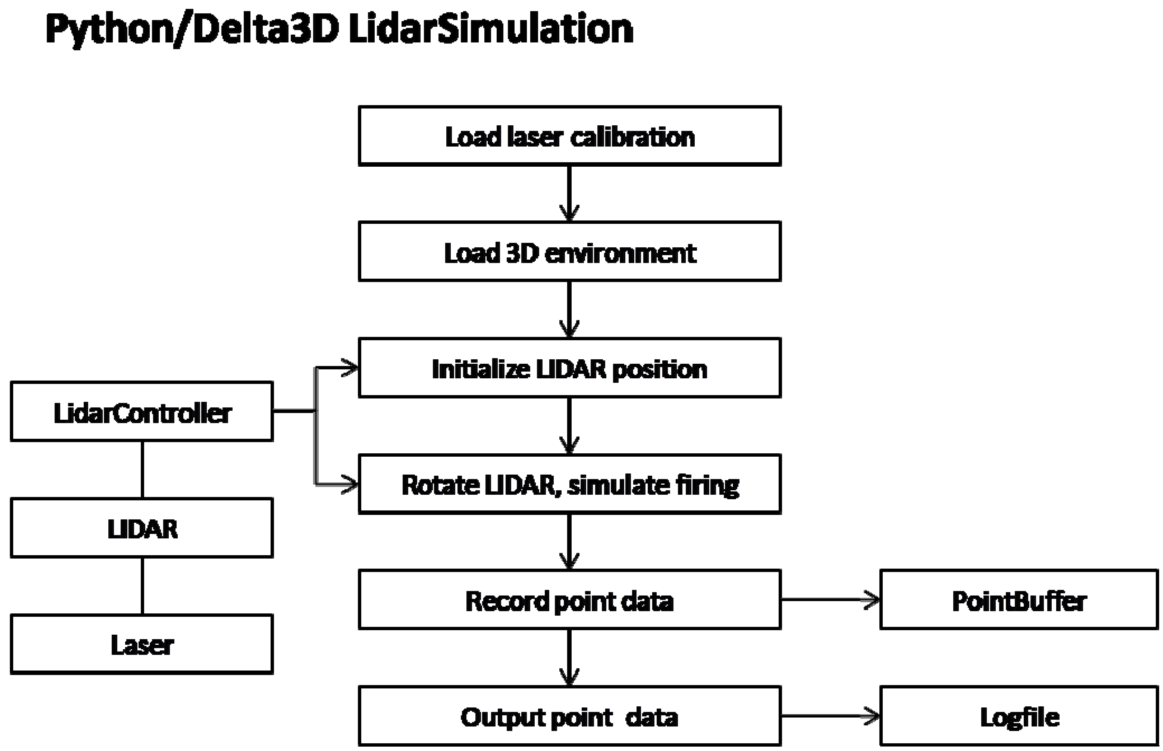


Figure 8. Python/Delta3D point cloud generator.
This figure depicts the processing flow for simulation of LIDAR point acquisition in a 3D graphical environment.

The Delta3D graphical simulation environment offers intersection testing functions that calculate ray intersection hit points from a given point to graphics objects in the simulated environment. Object labeling for each hit point is also available. This information allows the testing of correct segmentation and point labeling methods.

The steps in the process are summarized below.

- Load laser calibration: load the correction values for each laser.
- Load 3D environment: loads a Delta3D environment using 3D objects from *.osg* or *.ive* files.
- Initialize LIDAR position: initialize the LIDAR position and orientation in the world.

- Rotate LIDAR, simulate firing: the LIDAR object steps through rotation values while an intersection test is performed using a ray from the laser position to the first surface intersected in the graphical world.
- Record point data: record the Cartesian coordinates and range for each intersection test. Gaussian noise can be added at this step.
- Output point data: point data output to a file.
- LidarController: object that controls the position, orientation, and firing of the LIDAR object.
- LIDAR: object that represents the VLS, includes 64 laser objects.
- Laser: object representing a single laser, including the offset and correction values relative to the LIDAR origin.
- Point Buffer: data structure to contain all point intersection results.

a. LIDAR Object Implementation

The LIDAR object includes translation, orientation, and rotation speed variables. The 64 lasers are oriented relative to the LIDAR object position and orientation. The manufacturer's XML calibration file contains correction values for each laser and determines the simulated laser position and firing direction within the LIDAR object. Laser firings are simulated in sequence as the LIDAR object rotates in the graphical environment. Each laser firing range result is calculated through an intersection test along the direction of the laser firing. The first graphical object intersected determines the laser hit point and the resulting range value. A Gaussian distributed error is applied to the range value to simulate the error of the LIDAR system being modeled.

b. Three Dimensional Scene Implementation

A graphical scene implemented in Delta3D defines the simulated geometric features for testing. This graphical scene is loaded into a Delta3D environment from files in the *.osg* or *.ive* formats. These files can be created with 3D modeling software, such as Blender or constructed from source code. Figure 9 shows an example scene and the generated point cloud from the scene.

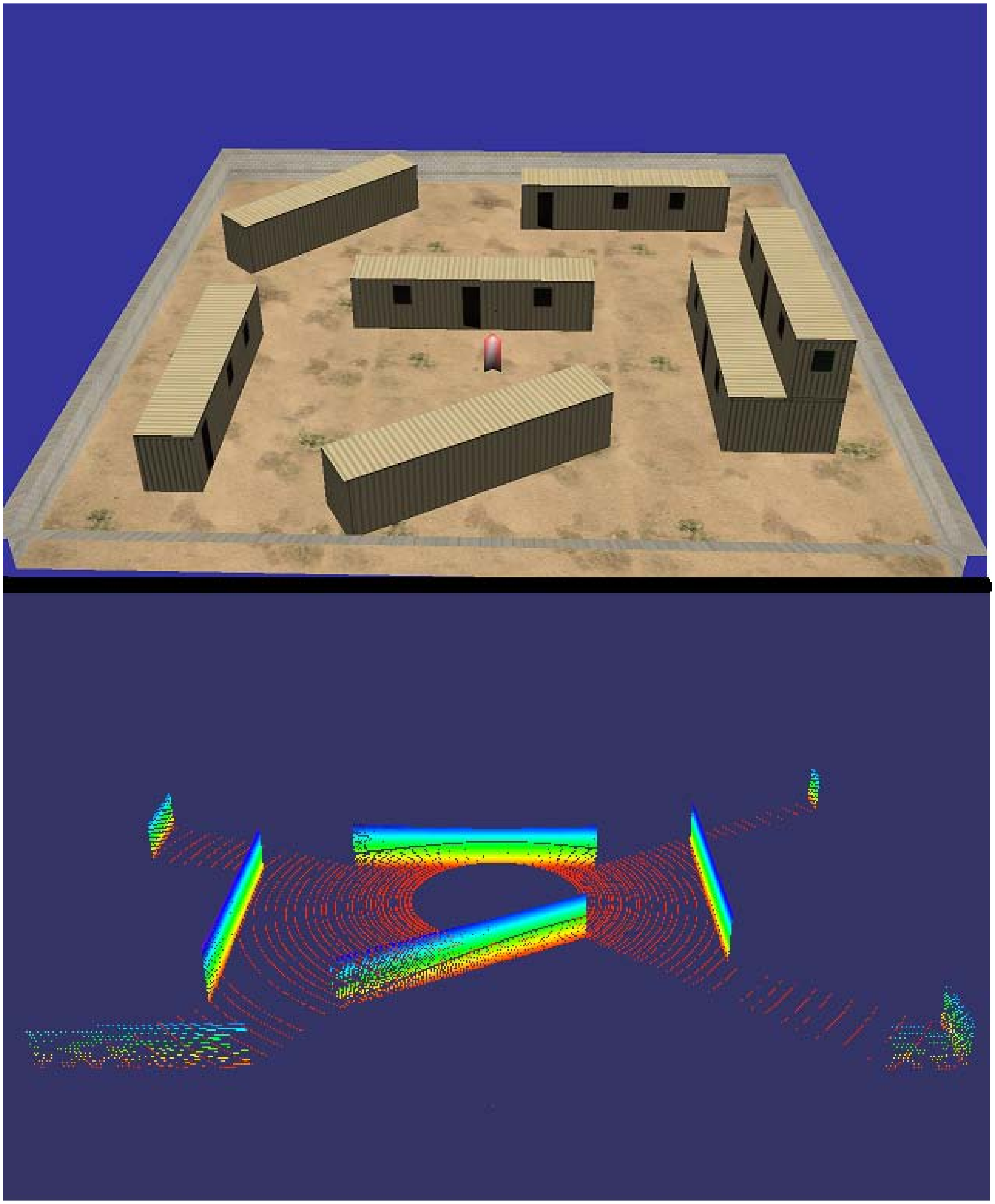


Figure 9. Delta3D graphical scene and LIDAR simulation point cloud. The red shape at the center of the Delta3D scene is the location of the LIDAR sensor. The rendered point colors indicate the increase in point elevation using a red to blue gradient.

c. Point Cloud Output

The output data formats available from the point cloud generator include a format that the Python simulation code can read, a comma separated value text file for processing in MATLAB, and a space delimited text file that can be read by the *LidarSimulation* offline processing code. The potential exists to extend the data output for UDP network packet transmission for a direct input to the online VLS data processing software. The current implementation in Python is slower than real-time due to the high number of intersection tests that need to be calculated per frame. For this reason, the simulation primarily stores a frame of data that is then accessed by the feature extraction code being tested.

C. OFFLINE VLS DATA PROCESSING

Implementing the fiducial detection method in an offline processing mode allows testing the data structures, fitting functions and overall effectiveness of the method with a greater level of control and inspection. The data structures and functions are the same as those in the online LIDAR processing code. An input log file is created from a live LIDAR packet stream, transformed from a recorded LIDAR packet stream, or generated with the Python Delta3D LIDAR simulation. The log file contains the range data organized by firing, packet, and frame. Figure 10 shows a flow chart for the offline VLS data processing software.

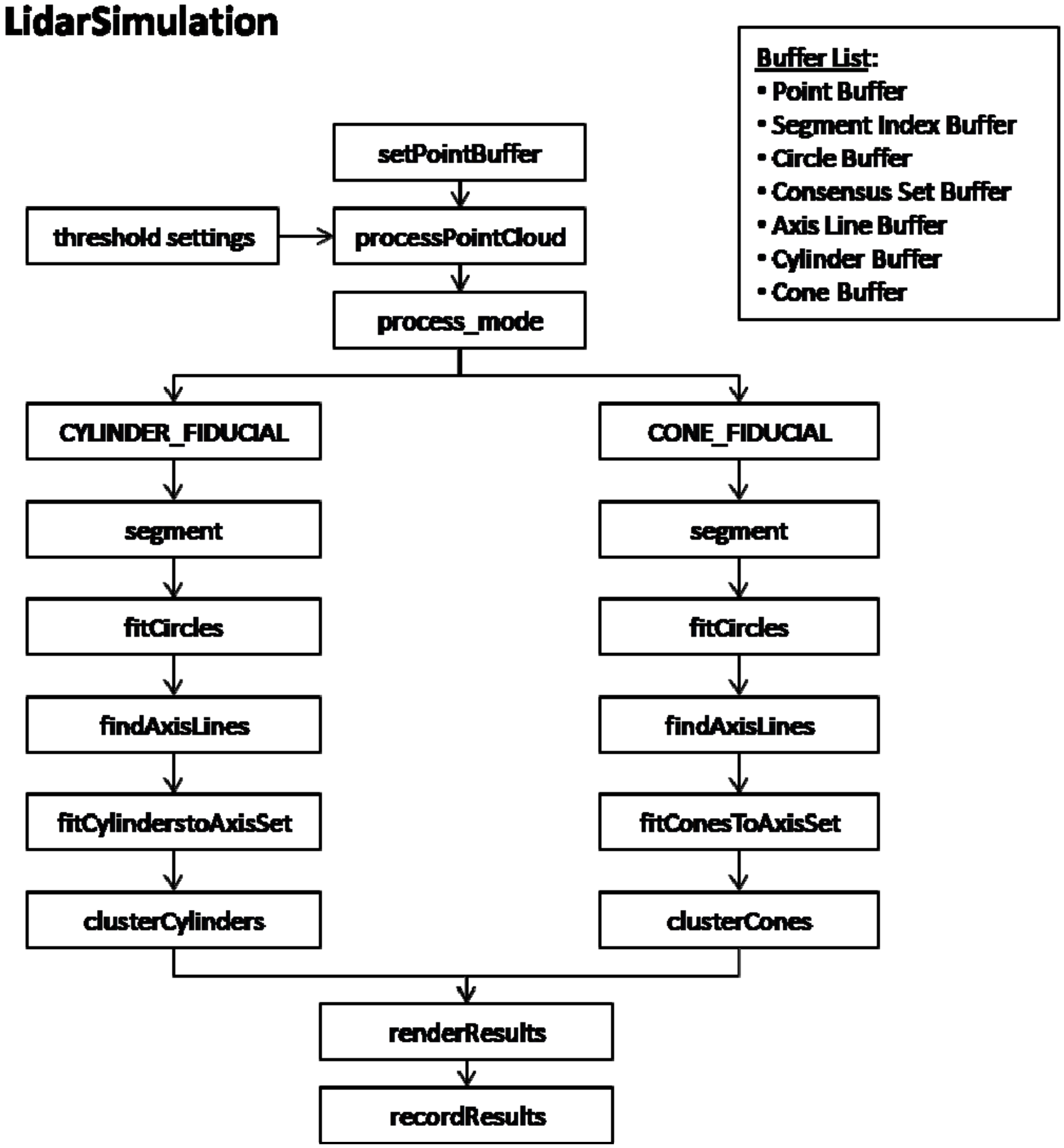


Figure 10. Offline VLS data processing.
This flow chart shows the offline processing of the LIDAR data.

The descriptions for each of the steps in Figure 11 are listed below.

- setPointBuffer: read point data for one frame to the point buffer.
- processPointCloud: process points based on the process option mode and threshold settings.

- CYLINDER_FIDUCIAL: mode to find cylinder fiducial.
- CONE_FIDUCIAL: mode to find cone fiducial.
- Segment: determine intervals of continuity and intervals of constant range filling the indices of the start of each segment in the *SegmentIndexBuffer*
- fitCircles: fit circles to the segments, filling *CircleBuffer* with fit arcs.
- findAxisLines: fit axis line hypotheses to the center points of the circle features to form consensus sets, use consensus set to fit an axis line hypothesis
- fitCylindersToAxisSet: fit cylinders to set of axis line hypotheses, fill CylinderBuffer
- fitConesToAxisSet: fit cones to set of axis line hypotheses, fill ConeBuffer
- clusterCylinders: combine co-located and similar cylinder hypotheses.
- clusterCones: combine co-located and similar cone hypotheses.
- recordResults: write results to a file
- renderResults: render the point, cylinder, cone buffers for visualization

D. ONLINE VLS DATA PROCESSING SOFTWARE

The online version of the VLS data processing software is designed for direct processing and classification of 3D point cloud data from the VLS in order to recognize a fiducial marker in the environment and estimate its pose. The same algorithms and data structures are used in the offline version of the VLS data processing software to allow testing and modification of the algorithms in either environment. Currently not every feature of the offline system is operational in the online version.

THIS PAGE INTENTIONALLY LEFT BLANK

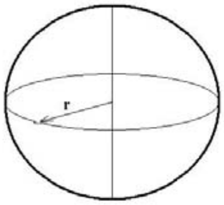
IV. EXPERIMENT

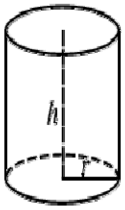
A. OVERVIEW

The following chapter describes an experimental setup to determine the performance of the proposed method for accuracy and precision of fiducial detections and the precision of the fiducial pose estimation. The experiment considers a mixed structure environment, varying the type of fiducial marker, the range between the LIDAR and fiducial marker, and the vertical angle between the LIDAR and the fiducial marker. The mixed structure environment includes several features that provide detection noise and false detection opportunities. The LIDAR height is changed to vary the angle of the lasers that scan the fiducial marker. The LIDAR system pose is not changed during an individual data capture event, but the range to the fiducial marker is changed between events. The variation in the LIDAR height and the fiducial range provides an indicator for how well the method is able to segment the data and fit fiducial marker features in static real world data. It is expected that similar results would be obtained from a dynamic experiment with relative motion between the sensor and fiducial marker.

B. FIDUCIAL MARKER CONSTRUCTION

Three test fiducial markers were constructed to collect data for testing the fiducial marker detection and pose estimation method: a cylinder, a cone, and a sphere. The geometric dimensions of these fiducial marker shapes are shown in Table 6.

Geometric Shape	**Dimensions**	**View**
Sphere	Radius = 10.25 inches (26 cm) Origin: center point.	

Cylinder	Height = 48 inches (121.9 cm)
	Radius = 6.25 inches (15.9 cm)
	Origin: center point of base.
	Orientation of axis = (0,0,1)
		

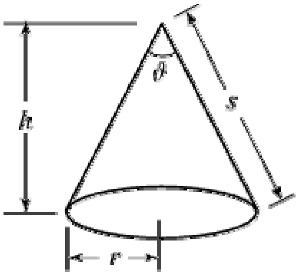
Cone	Base = 31 inch radius (78.7 cm) Origin: center of base of cone. Height = 60 inches (152.4 cm) Angle of apex = 56 degrees Angle of base = 62 degrees Orientation of axis = (0,0,1)	

Table 9. Fiducial marker dimensions.

The cylinder is a prefabricated concrete construction form made of fiberboard. The sphere is an inflatable exercise ball. The cone is constructed of a plywood frame covered with paperboard measured to the dimensions of a half-cone. Figure 11 depicts the fiducial markers as they were used in the experimental test of the processing method.

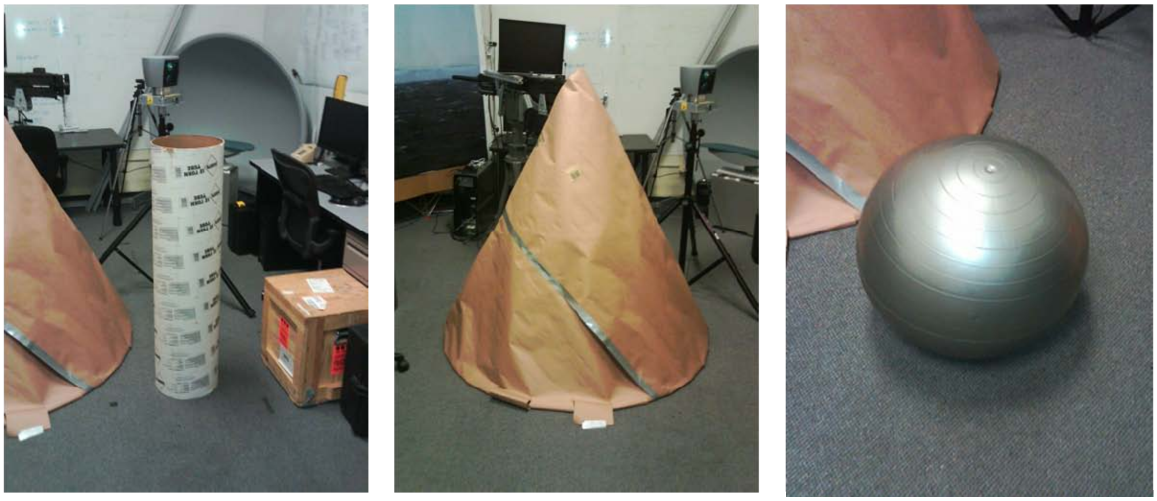


Figure 11. Photos of fiducial markers.
These photos depict the three fiducial markers as used in the experimental data collection.

C. EXPERIMENT SETUP

The experiment captures sample test data for analysis using the Velodyine HDL-64E S2 LIDAR system. Fiducial markers of cylindrical, conical, and spherical geometries are used as targets in an open sidewalk area between two buildings. Other environment features include square concrete pillars, round garbage cans, trees, bushes, and signposts.

1. Design of Experiment

The three factors this experiment addresses are the fiducial marker shape, the fiducial marker range, and the vertical difference between the LIDAR and the fiducial marker. The experiment layout consists of the LIDAR mounted on a tripod with the fiducial markers placed in fixed positions on the ground. Approximate ground truth data is obtained through measurements with a laser level and laser range finder to identify the relative location of each fiducial marker from the LIDAR coordinate system origin. The base of each fiducial marker rests on the ground. Center lines marked on the fiducial markers assist alignment in the x and y directions of the horizontal plane. The locations are measured for horizontal distance and an elevation offset from the LIDAR origin. The laser level provides ground point distance and elevation from the LIDAR. The range is varied from five to 30.0 meters. No orientation measurement is obtained of the fiducial axis offline, but the base of the LIDAR is leveled to align the vertical axis perpendicular to the ground. The LIDAR rotates at five Hz speed to provide a higher sampling rate across each laser scan line. To record test data, the LIDAR UDP packets are captured using a laptop computer connected to the LIDAR via an Ethernet cable. The test data are analyzed offline using the packet captures. Each recorded set of packets allows playback at real-time speeds over the same UDP network as the live system. A summary of measurements of the experimental layout is given in Table 10.

Range (meters)	**Level measurement (meters)**	**Ground z-axis location (meters) LIDAR at 1.225 m**	**Ground z-axis location (meters) LIDAR at 2.00 m**
5	0.047	127.2625	2.048
10	0.083	130.755	2.083
15	0.108	133.295	2.108
20	0.092	131.7075	2.092
25	0.052	127.739	2.052
30	0.035	125.9925	2.036
LIDAR location measurements. Base Height run 1 = 1.225 m Base Height run 2 = 2.000 m Rotation Speed = 5 Hz = 300 RPM			

Table 10. Experimental layout dimensions.

D. DATA ANALYSIS

The recorded data consists of a series of 10 to 14 second data captures of the fiducial markers in a sidewalk area between two buildings. There are several tree, vehicle, and concrete sidewalk features that present noise and false detection opportunities. Each test data capture is performed with the LIDAR in a measured position on the ground relative to the LIDAR sensor. The following procedure is used to evaluate the fiducial marker detection and pose estimation method.

- A total of 60 +/- 10 revolutions of the sensor are recorded at a rate of five Hz.
- The detections are classified as true positive, false positive and false negative. A true positive classification is given to fiducial marker estimates in positions within one meter of the true position. A false positive detection occurs when there is a fiducial detection outside of the one meter threshold. False negative detections are labeled when there is no detection but the fiducial is truly present.
- A ground truth estimate is calculated from the mean value of five true positive classified fiducial marker estimates. The sample mean, sample standard deviation, and standard error are calculated.
- For the remaining true positive detections, the estimate is adjusted by the ground truth sample. The resulting adjusted mean error is calculated as the absolute difference between the estimate value and the ground truth value.

Offline processing and analysis are used on the recorded data to analyze accuracy and precision.

1. Fiducial Marker Detection Accuracy and Precision

Analysis of fiducial marker detection accuracy measures the instances of true positive, false positive, and false negative detections. A true positive detection occurs when the method detects a fiducial marker of the correct dimensions in the correct position in the environment where one exists. A false negative detection happens when the method fails to detect a fiducial marker when there is one in the environment. A false positive is when the method detects a fiducial marker in the environment in a location in which there is none in the environment.

Fiducial marker detection accuracy, precision, and recall are calculated as

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

2. Fiducial Marker Pose Estimation Precision

The precision of pose estimation is analyzed for those instances of true positive detections. The error of the pose estimation is compared to a sampled ground truth value. The LIDAR range data is more precise than common methods for measurement of true fiducial marker pose. For this reason, a sample of true positive detections is used to derive a mean pose estimate that is used as a ground truth value. The system pose estimation values are then compared for variation around this mean ground truth value.

Fiducial pose estimation performance is quantified by the deviation of the fiducial marker pose estimated by the system and the ground truth fiducial marker pose.

Where x_i is a sampled measurement, such as the radius of a detected cylinder, the mean of sample values $\overline{x}$ over multiple data frames n is calculated as

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

The measurement is compared to a ground truth value to determine the error ε_i where

$$\varepsilon_i = x_i - \overline{x}$$

The standard deviation of errors over multiple data frames is defined as

$$s = \sqrt{\frac{\sum_{i=1}^{n} {\varepsilon_i}^2}{n-1}}$$

The standard error of the mean is defined as

$$SE = \frac{s}{\sqrt{n}}$$

The mean square of estimate errors x_i over multiple data frames is defined as

$$MSE = \frac{\sum_{i=1}^{n} \varepsilon_i^{\,2}}{n-1}$$

The fiducial marker position estimation error for each frame of data is calculated with a root mean square error (RMSE) value, where n is the number of data points, (x_e, y_e, z_e) are the estimated coordinates, and (x_g, y_g, z_g) are the ground truth coordinates. The position estimation is evaluated using the Euclidean range estimate in the xy-plane and the vertical axis location estimate relative to the LIDAR. The Euclidean range estimate in the xy-plane is $\rho_e = \sqrt{x_e^{\,2} + y_e^{\,2}}$ and the ground truth range is $\rho_g = \sqrt{x_g^{\,2} + y_g^{\,2}}$. The root mean square error for the range estimate is computed where

$$RMSE_\rho = \sqrt{\frac{\sum_{i=1}^{n} (\rho_e - \rho_{true})^2}{n}}$$

The fiducial marker vertical location is estimated as z_e and the root mean square error for the vertical position estimate is

$$RMSE_z = \sqrt{\frac{\sum_{i=1}^{n} (z_e - z_g)^2}{n}}$$

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

A. EXPERIMENTAL RESULTS

The results of the fiducial detection and estimation method on the experimental test data are presented in this chapter. The objective of the experimental analysis is to characterize the laser data for fiducial detection accuracy and precision. Preliminary test runs determine suitable ranges for thresholds. Initial analysis also reveals that the sphere fiducial marker used in the experiment is not large enough to be accurately detected with the LIDAR device and methods presented here. Not enough laser scan lines intersect the sphere. Hence, the results are analyzed for the cylinder and cone fiducial markers.

Table 11 shows the threshold settings used for the data analysis shown in this chapter. All thresholds are held constant for all test events with one exception. The cylinder fiducial was not detected consistently beyond 20 meters. This is due to the fewer number of intersecting laser scan lines and the fewer number of arcs used to make a cylinder hypothesis. The minimum number of arcs required to form a hypothesis cylinder or cone are listed below.

- At 10 meters a minimum of five arcs
- At 15 meters a minimum of five arcs
- At 20 meters a minimum of four arcs
- At 25 meters a minimum of four arcs
- At 30 meters a minimum of three arcs

The lower minimum number of arcs required to form a hypothesis facilitates detection of the fiducial markers, but increases the computational complexity of the search as there are more combinations of arcs to search for candidate hypotheses. This same threshold is varied in the same way for cone fiducial detection events in order to make a direct comparison between cone and cylinder detection with the same threshold settings.

Threshold	**Value**	**Description**
N_p	2	Initial set of points to define an axis hypothesis
$N_{arcs-cone}$	3,4, or 5	Minimum number of arcs for cone hypothesis
$N_{arcs-cylinder}$	3, 4, or 5	Minimum number of arcs for cylinder hypothesis
$e_{cylinder}$	0.05 m	RMSE tolerance for point fit to a cylinder
e_{cone}	0.10 m	RMSE tolerance for point fit to a cone
e_{circle}	0.01 m	RMSE tolerance for point fit to an arc
h_{cone}	3.0 m	Maximum cone height threshold to eliminate tall thin cones that fit to arcs on a cylinder
$r_{\max}$	0.1 m	Maximum radius of arcs considered for cylinder or cone fit
$r_{\min}$	0.5 m	Minimum radius of arcs considered for cylinder or cone fit
t_{rot}	5 deg	Rotation value to cluster neighbor circle arcs
t_{line}	0.1 m	Maximum distance arc center point to an axis line
$t_{orientation}$	0.1	Cosine of difference angle between axis lines
t_{aspect}	0.3	Maximum difference in cone aspect ratio, $\frac{height}{base}$
$t_{horizontal\_gradient}$	0.005 m	Horizontal gradient threshold for constant range flag
$t_{segment}$	0.01 m	Gradient threshold for segment divisions

Table 11. Experimental threshold settings.

Changing the minimum and maximum radius thresholds to a small tolerance around the expected fiducial marker radius value eliminates arcs from the list to be searched for candidate fiducial hypotheses. In the case of the cylinder this threshold can effectively limit the number of hypotheses searched. For cone fiducial marker detection with this method, a wider range of radii need to be considered in order to locate the arcs that are fit to the cone surface.

1. Segmentation and Feature Fitting

The results of the segmentation process in Table 12 show the mean count of points, segments, arcs, axis lines, cylinders and cones extracted from the range data. A consistent number of points and segments for each event is evident. The number of points includes both valid and invalid range values from the sensor. The invalid points are culled from the point data and only valid range data are considered for further processing. The number of arcs fit is dependent on the threshold setting for root mean square error for the fit of data points to a circular arc. With a greater allowable error a larger number of arc features is found.

The number of arcs, axis lines, and cylinder or cone features fit decreases with the range of the fiducial marker. When the range of the fiducial is increased, there are fewer surface data points obtained with a corresponding decrease in the number of arcs and axis lines found. The lower number of arcs and axis lines available decreases the overall fit of the data points to the hypothesized fiducial model’s surface. With fewer arcs found, there is a less certain fit of the axis line to construct the fiducial hypothesis model. Even arc hypotheses that correspond to the true fiducial marker stand a smaller chance of contributing to the marker detection if they exhibit a high error of fit.

Other features in the environment may return a false positive detection when their feature parameters are within the threshold limits for the fiducial, and have a lower error of fit to the data than the true fiducial marker. Adjusting the threshold settings to more closely match the fiducial marker of interest eliminates many of the features not associated with the marker. The drawback is that fewer features are available to construct cone or cylinder hypotheses.

Fiducial Type	LIDAR height (m)	Range (m)	Mean Counts Per Frame						
			pts	segs	arcs	axis lines	cylinders	cones	clusters
cylinder	1.22	10.0	276,480	49,065	504.0	41.4	35.0	-	6.7
		15.0	276,480	49,104	519.7	32.9	28.2	-	7.0
		20.0	276,480	48,921	504.8	38.8	30.3	-	11.0
		25.0	276,480	49,002	499.8	34.5	25.9	-	10.0
		30.0	276,480	48,943	514.4	83.0	56.5	-	32.5
	2.00	10.0	276,480	34,037	729.2	33.3	24.5	-	7.9
		15.0	276,480	33,781	743.9	31.6	22.9	-	7.3
		20.0	276,480	33,689	738.0	24.8	15.8	-	7.4
		25.0	276,480	33,614	726.4	62.7	31.3	-	18.3
		30.0	276,480	33,789	755.0	196.7	99.5	-	68.0
cone	1.22	10.0	276,480	49,622	1,483.9	389.0	-	17.3	6.3
		15.0	276,480	49,040	1,432.9	323.8	-	10.7	4.4
		20.0	276,480	48,849	1,446.3	595.0	-	11.8	7.8
		25.0	276,480	48,973	1,592.4	921.2	-	19.1	14.9
		30.0	276,480	51,451	1,570.2	836.3	-	15.5	12.1
	2.00	10.0	276,480	33,688	2,053.1	648.7	-	11.4	2.9
		15.0	276,480	33,204	2,018.1	601.7	-	6.5	3.3
		20.0	276,480	33,141	2,036.5	1,111.3	-	7.2	5.1
		25.0	276,480	33,409	2,094.7	11.44.1	-	9.2	6.8
		30.0	276,480	33,257	2,051.9	1,060.3	-	6.7	5.0

Table 12. Segmentation and feature fitting results.

2. Arc Radius Estimation Bias

The cylinder radius estimation exhibits an error in radius estimation that affects the overall detection performance of the method. This radius estimation bias indicates an overestimation of the radius of the arcs on the fiducial cylinder that increases with range. The observed approximate range overestimation is shown in Table 13. Since the fiducial marker detection relies on excluding hypotheses with radii not similar to the radius of the fiducial, adjusting the fiducial marker radius estimate by these values improves the detection of cylinder fiducial markers. Figure 12 depicts the adjusted mean error of cylinder radius estimation when the radius estimate bias is included.

There is a near constant radius estimate error for the cylinder fiducial marker. The precision of the radius estimate is on the order of one to two centimeters. The VLS system used in the experiment has a reported range error of about 1.5 centimeters matching the observed values (HDL-64E S2 Users Manual).

Fiducial Type	LIDAR Height (m)	Range (m)	Radius bias (m)
Cylinder	1.22	10.0	+0.03
		15.0	+0.03
		20.0	+0.05
		25.0	+0.06
		30.0	+0.06
	2.00	10.0	+0.01
		15.0	+0.02
		20.0	+0.05
		25.0	+0.06
		30.0	+0.07

Table 13. Cylinder arc radius bias.

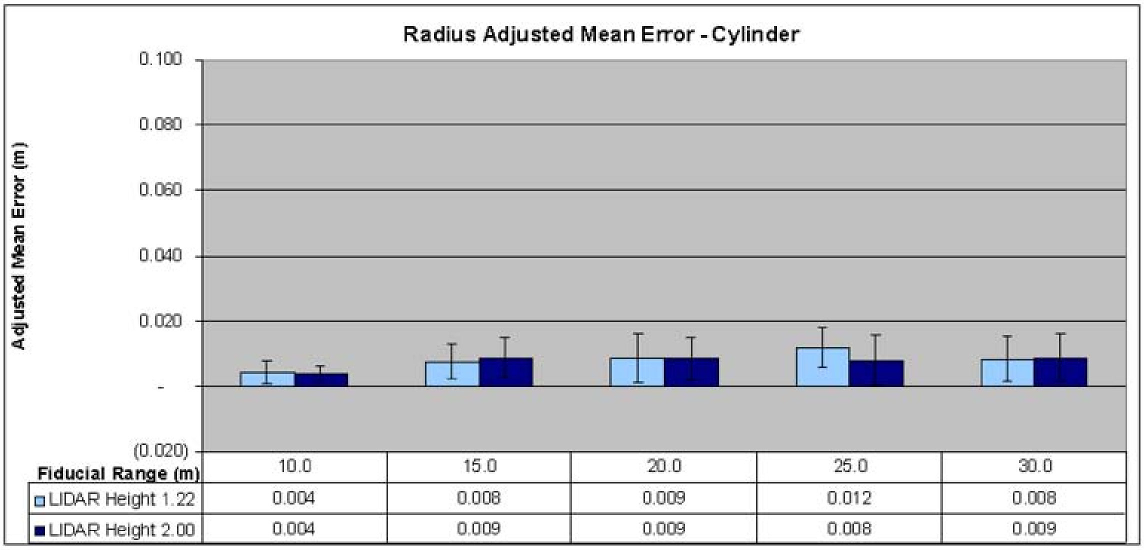
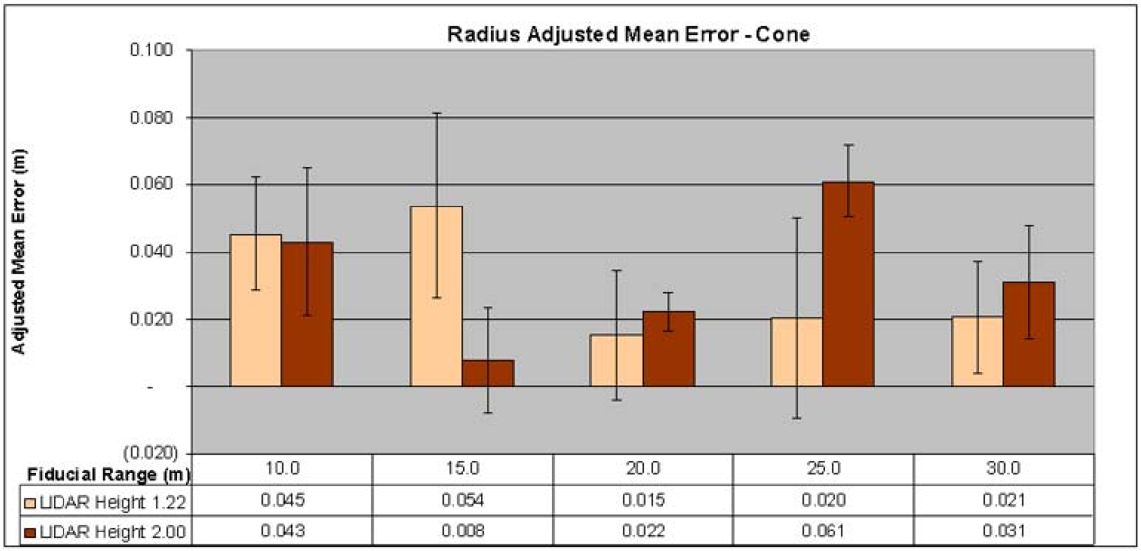


Figure 12. Radius adjusted mean error – cylinder.

The cone base radius estimate is more inaccurate than for the cylinder. Figure 13 shows the wide variance in base radius estimation for the cone. Although this variance is greater than for the cylinder, in the case of estimating cone fiducial marker pose the base radius estimate is not needed. The estimate of the location of the apex of the cone is the most critical.



Fiducial Range (m)	10.0	15.0	20.0	25.0	30.0
□ LIDAR Height 1.22	0.045	0.054	0.015	0.020	0.021
■ LIDAR Height 2.00	0.043	0.008	0.022	0.061	0.031

Figure 13. Base radius adjusted mean error – cone.
The cone base radius estimate is calculated from the hypothesis cone. A cone with the same apex angle as the fiducial marker of interest, but with a different height will result in a different base radius.

3. Fiducial Detection Accuracy

The accuracy and precision for fiducial detection is calculated using the prior knowledge of the position of the fiducial in the test environment and the radius estimate bias from Table 13. The fiducial detection accuracy, precision, and recall are given in Table 14. These results indicate a decrease in fiducial detection accuracy and precision with range. Table 15 gives the mean number of arcs, mean number of points, and root mean square error for the true positive classifications of each test event. For each test event shown in Table 14, all true positive detections are used to calculate the mean and standard deviation of the number of point and arc features. The results are shown in Table 15 along with the average root mean square error of the point fit to the fiducial geometry. With increasing range, there is a decrease in the number of points and arcs available to determine the fiducial marker location. These decreases correspond to the range dependent decrease in the number of laser scan lines intersecting the marker.

The cone at 25 meters with the LIDAR at 1.22 meters is lower than anticipated. Viewing the data shows that there was some sort of object in the environment that consistently had a better fit to the data than the true fiducial marker.

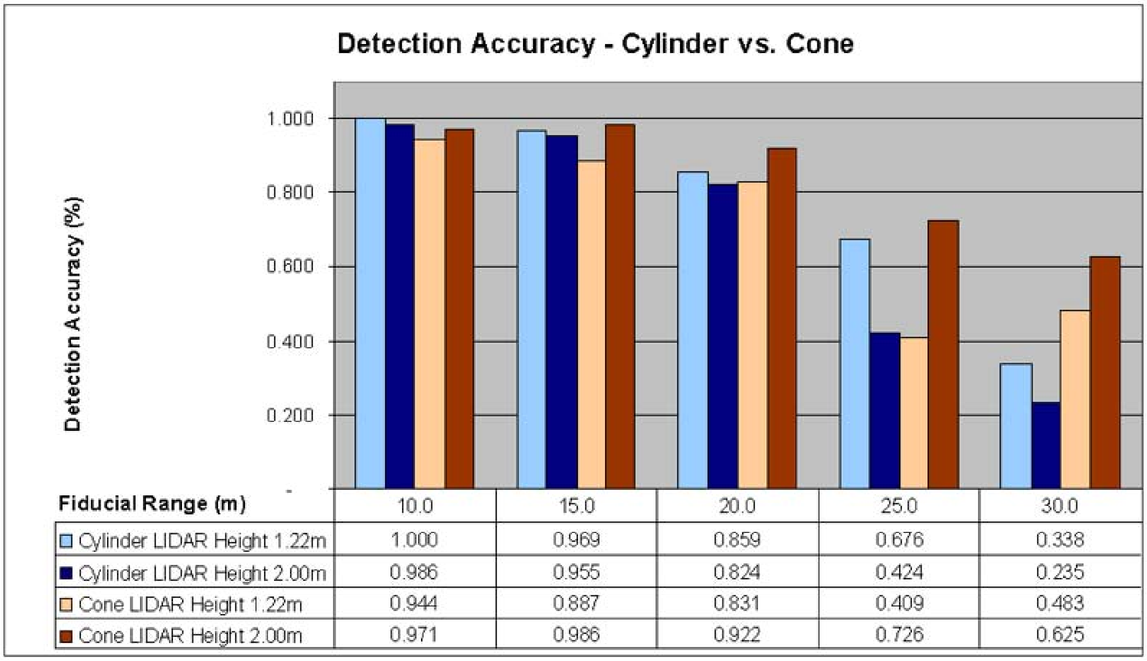
Fiducial Type	LIDAR height (m)	Range (m)	Detection			Accuracy	Precision	Recall
			TP	FP	FN			
Cylinder	1.22	10.0	70	0	0	1.00	1.00	1.00
		15.0	62	2	0	0.97	0.97	1.00
		20.0	61	10	0	0.86	0.86	1.00
		25.0	48	23	0	0.68	0.68	1.00
		30.0	24	47	0	0.34	0.34	1.00
	2.00	10.0	70	1	0	0.99	0.99	1.00
		15.0	63	3	0	0.95	0.95	1.00
		20.0	56	12	0	0.82	0.82	1.00
		25.0	28	38	0	0.42	0.42	1.00
		30.0	16	52	0	0.24	0.24	1.00
Cone	1.22	10.0	67	4	0	0.94	0.94	1.00
		15.0	63	7	1	0.89	0.90	0.98
		20.0	59	10	2	0.83	0.86	0.97
		25.0	25	29	0	0.46	0.46	1.00
		30.0	28	28	2	0.48	0.50	0.93
	2.00	10.0	67	1	1	0.97	0.99	0.99
		15.0	70	1	0	0.99	0.99	1.00
		20.0	59	5	0	0.92	0.92	1.00
		25.0	45	13	4	0.73	0.78	0.92
		30.0	40	2	22	0.63	0.95	0.65

Table 14. Fiducial detection accuracy, precision, and recall.
This table gives the True Positive (TP), False Positive (FP), and False Negative (FN) counts per frame of LIDAR data. Accuracy, precision, and recall are calculated as described in Chapter IV.

Fiducial Type	LIDAR height (m)	Range (m)	Mean # arcs	Mean # pts	Mean RMS Error	Std Dev # arcs	Std Dev # pts	Std Dev RMS Error
Cylinder	1.22	10.0	7.4	339.0	0.017	2.14	97.17	0.004
		15.0	6.8	205.5	0.019	1.64	50.90	0.003
		20.0	4.6	104.1	0.024	0.85	19.00	0.007
		25.0	4.6	85.7	0.028	0.82	16.26	0.007
		30.0	3.6	54.3	0.030	0.92	13.89	0.007
	2.00	10.0	8.2	371.3	0.019	1.89	86.56	0.001
		15.0	6.6	200.5	0.017	1.41	42.94	0.003
		20.0	5.4	126.9	0.019	0.59	15.38	0.008
		25.0	4.2	77.7	0.023	0.39	7.73	0.007
		30.0	3.1	47.4	0.017	0.34	5.50	0.006
Cone	1.22	10.0	7.7	536.6	0.047	1.46	144.15	0.014
		15.0	6.4	342.6	0.035	1.25	69.46	0.009
		20.0	5.3	228.2	0.031	0.75	30.46	0.004
		25.0	3.8	122.6	0.036	0.71	27.47	0.013
		30.0	3.3	89.9	0.039	0.44	11.80	0.013
	2.00	10.0	8.7	680.6	0.050	1.76	183.91	0.008
		15.0	6.8	406.9	0.037	0.67	38.79	0.006
		20.0	4.3	180.3	0.031	0.54	17.27	0.007
		25.0	4.2	130.3	0.032	0.40	19.10	0.001
		30.0	4.1	113.0	0.042	0.30	18.28	0.012

Table 15. Fiducial root mean square error of fit to data points.
The mean number of arcs, data points, and root mean square error of fit of the data are shown for each true positive fiducial marker detection.

Figure 14 gives the detection accuracy of the cylinder and cone with respect to fiducial marker range from the LIDAR sensor. For both the cylinder and the cone, the detection accuracy decreases with increasing range from the LIDAR sensor. Overall the cone maintains better detection out to 30 meters. The less than expected value for the cone at 25 meters is approximately the same for the cylinder.

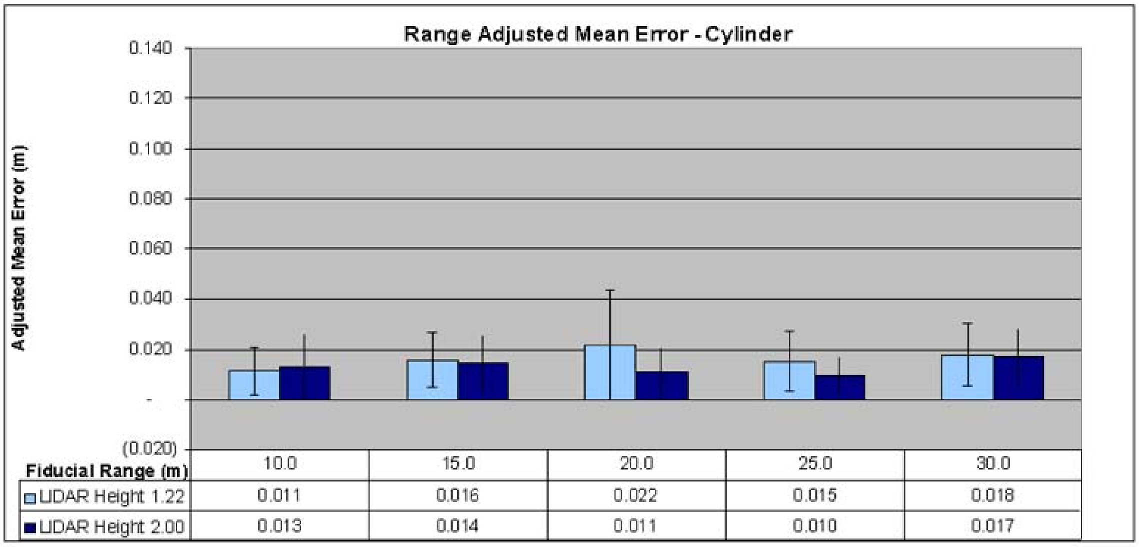


Fiducial Range (m)	10.0	15.0	20.0	25.0	30.0
□ Cylinder LIDAR Height 1.22m	1.000	0.969	0.859	0.676	0.338
■ Cylinder LIDAR Height 2.00m	0.986	0.955	0.824	0.424	0.235
□ Cone LIDAR Height 1.22m	0.944	0.887	0.831	0.409	0.483
■ Cone LIDAR Height 2.00m	0.971	0.986	0.922	0.726	0.625

Figure 14. Cylinder vs. cone fiducial detection accuracy.

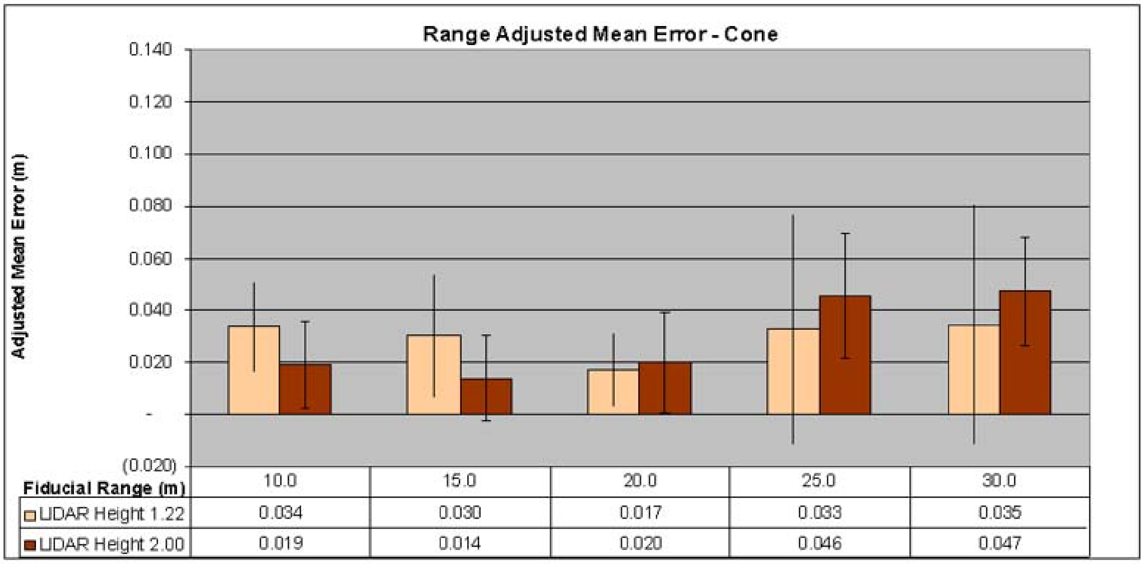
4. Fiducial Position Estimation Error

Fiducial position estimation error is evaluated based on the precision of the estimate. This precision is calculated using a subset of the true positive detections for a test event; an adjusted mean error and standard deviation are computed for the remaining true positive detections for that event. The range errors are given in Figure 15 and Figure 16 with the numerical mean results shown in Table 16 and the standard deviations given in Table 17.



Fiducial Range (m)	10.0	15.0	20.0	25.0	30.0
□ LIDAR Height 1.22	0.011	0.016	0.022	0.015	0.018
■ LIDAR Height 2.00	0.013	0.014	0.011	0.010	0.017

Figure 15. Adjusted range mean error – cylinder.



Fiducial Range (m)	10.0	15.0	20.0	25.0	30.0
□ LIDAR Height 1.22	0.034	0.030	0.017	0.033	0.035
■ LIDAR Height 2.00	0.019	0.014	0.020	0.046	0.047

Figure 16. Adjusted range mean error – cone.

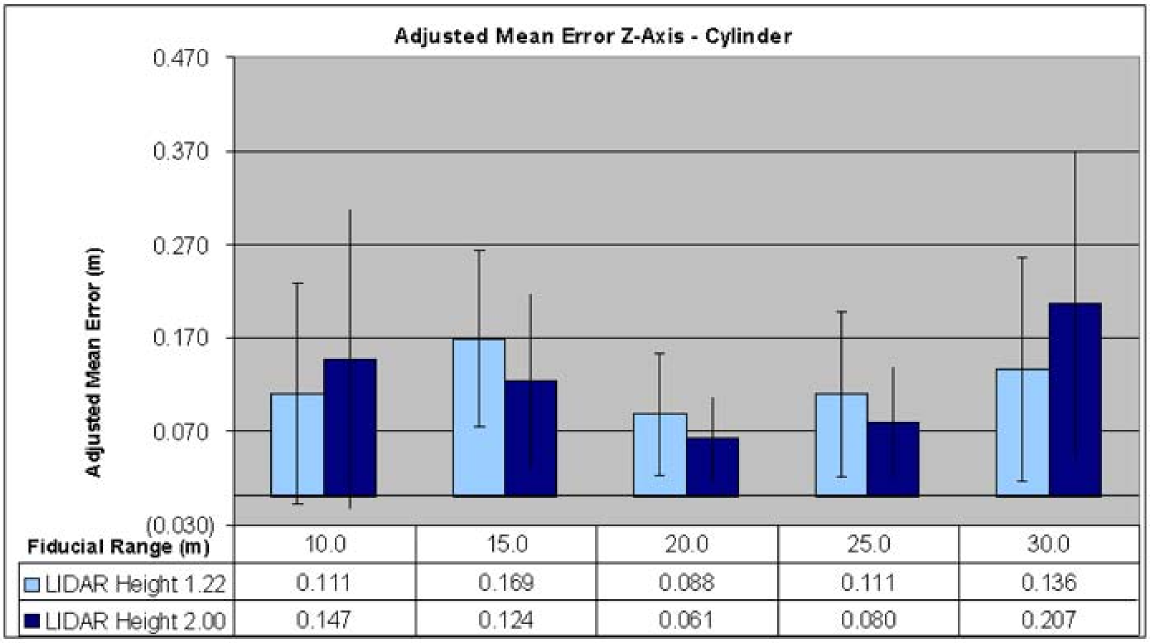
Fiducial Type	LIDAR height (m)	Range (m)	Adjusted Position Mean Error (m)				
			x	y	z	Radius	Range
Cylinder	1.22	10.0	0.002	0.012	0.115	0.004	0.012
		15.0	0.005	0.014	0.143	0.007	0.014
		20.0	0.004	0.021	0.087	0.009	0.021
		25.0	0.004	0.015	0.109	0.012	0.015
		30.0	0.004	0.019	0.104	0.010	0.019
	2.00	10.0	0.003	0.013	0.147	0.004	0.013
		15.0	0.002	0.015	0.117	0.010	0.015
		20.0	0.003	0.012	0.063	0.011	0.012
		25.0	0.004	0.009	0.087	0.009	0.009
		30.0	0.005	0.017	0.229	0.009	0.017
Cone	1.22	10.0	0.009	0.037	0.029	0.046	0.037
		15.0	0.007	0.033	0.017	0.061	0.033
		20.0	0.006	0.015	0.022	0.018	0.015
		25.0	0.012	0.089	0.033	0.038	0.089
		30.0	0.008	0.034	0.029	0.021	0.034
	2.00	10.0	0.005	0.019	0.022	0.043	0.019
		15.0	0.004	0.015	0.010	0.008	0.015
		20.0	0.005	0.021	0.026	0.028	0.021
		25.0	0.005	0.046	0.020	0.061	0.046
		30.0	0.013	0.044	0.038	0.032	0.044

Table 16. Adjusted position mean error.

Fiducial Type	LIDAR height (m)	Range (m)	Adjusted Position Standard Deviation Error				
			x	y	z	Radius	Range
Cylinder	1.22	10.0	0.002	0.010	0.113	0.004	0.010
		15.0	0.004	0.011	0.100	0.005	0.011
		20.0	0.003	0.022	0.063	0.007	0.022
		25.0	0.002	0.012	0.088	0.007	0.012
		30.0	0.003	0.013	0.116	0.007	0.013
	2.00	10.0	0.003	0.014	0.161	0.002	0.013
		15.0	0.002	0.012	0.097	0.007	0.012
		20.0	0.001	0.010	0.046	0.007	0.010
		25.0	0.003	0.008	0.062	0.007	0.008
		30.0	0.003	0.011	0.227	0.008	0.011
Cone	1.22	10.0	0.006	0.018	0.026	0.016	0.018
		15.0	0.005	0.024	0.015	0.039	0.024
		20.0	0.006	0.014	0.017	0.020	0.014
		25.0	0.006	0.030	0.012	0.016	0.030
		30.0	0.006	0.045	0.017	0.017	0.045
	2.00	10.0	0.005	0.017	0.016	0.022	0.017
		15.0	0.005	0.016	0.012	0.016	0.016
		20.0	0.004	0.018	0.016	0.006	0.018
		25.0	0.006	0.024	0.022	0.011	0.024
		30.0	0.006	0.037	0.022	0.018	0.036

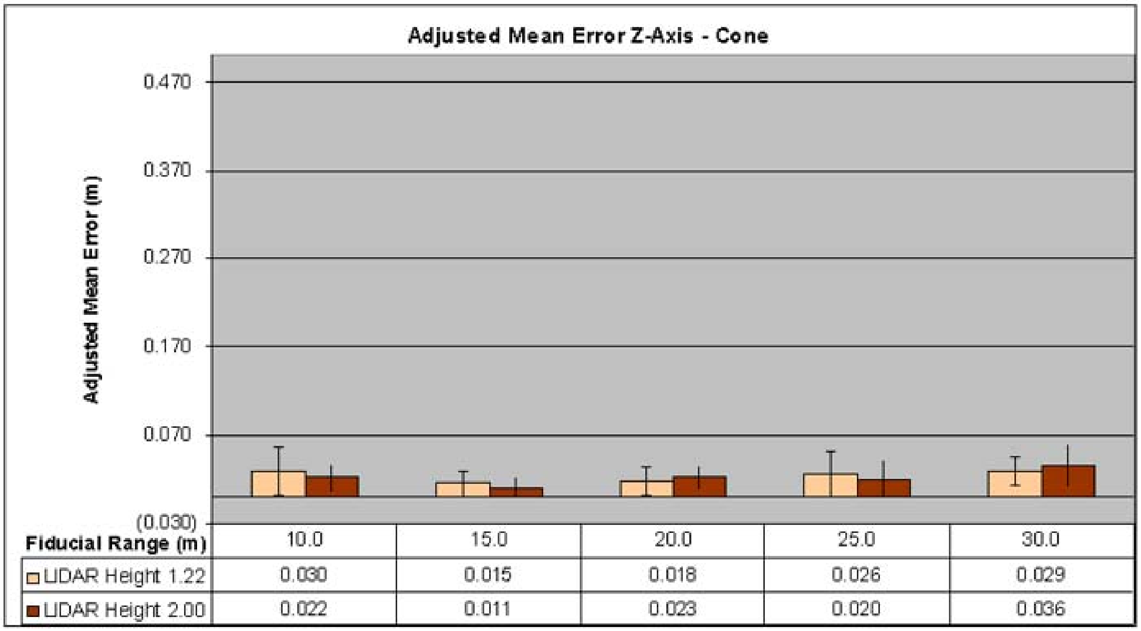
Table 17. Adjusted position error standard deviation.

The vertical position estimation errors are given in Figure 17 and Figure 18. These results show the cylinder maintains a relatively small range estimate error and a larger estimate error in the z-axis. The vertical position error is smaller for the cone, but the range error was greater.



Fiducial Range (m)	10.0	15.0	20.0	25.0	30.0
□ LIDAR Height 1.22	0.111	0.169	0.088	0.111	0.136
■ LIDAR Height 2.00	0.147	0.124	0.061	0.080	0.207

Figure 17. Adjusted mean error for Z-axis position estimate – cylinder.



Fiducial Range (m)	10.0	15.0	20.0	25.0	30.0
□ LIDAR Height 1.22	0.030	0.015	0.018	0.026	0.029
■ LIDAR Height 2.00	0.022	0.011	0.023	0.020	0.036

Figure 18. Adjusted mean error for Z-axis position estimate – cone.

5. Fiducial Orientation Estimation Error

The experiment data collected does include a measurement for the fiducial marker axis orientation, but the LIDAR sensor vertical axis is aligned perpendicular to the ground with a carpenter's level. The orientation of each detected fiducial marker is compared to the LIDAR axis orientation. Adjusted orientation values are computed for true positive detections using the same method as for the position values. The orientation differences are then transformed into the number of degrees of difference between the two axes. The adjusted mean difference angle between the estimated fiducial orientation and the approximate ground truth orientation are shown in Table 18.

Fiducial Type	LIDAR height (m)	Range (m)	Adjusted Axis Angle Error (deg)	
			Mean	Std Dev
Cylinder	1.22	10.0	1.6	1.2
		15.0	2.5	1.3
		20.0	3.9	6.2
		25.0	2.7	1.8
		30.0	3.2	2.1
	2.00	10.0	2.4	1.2
		15.0	1.5	1.0
		20.0	2.0	1.5
		25.0	1.3	1.0
		30.0	3.2	2.2
Cone	1.22	10.0	2.0	0.8
		15.0	2.4	1.2
		20.0	1.2	1.2
		25.0	6.2	3.1
		30.0	2.8	2.0
	2.00	10.0	1.8	1.5
		15.0	1.3	1.3
		20.0	2.0	1.7
		25.0	1.8	1.2
		30.0	1.3	1.0

Table 18. Adjusted axis angle error.

VI. DISCUSSION

A. EXPERIMENTAL RESULTS

The fiducial marker detection and pose estimation software was applied to the test data and the resulting fiducial marker pose estimates were investigated. In general, the threshold settings determine the system detection criteria and its ability to make accurate fiducial marker pose estimates. The experiment holds the threshold settings constant for all experimental trials except for the number of arcs needed to form an axis line hypothesis. This number was varied with range such that cylinder fiducials were able to be detected at greater ranges.

1. Pose Estimate

The pose estimation results in Figure 17 and Figure 18 show a larger mean error in the z-axis vertical position than for the range. This is due to the variance in the number of laser scan lines that intersect the fiducial marker. The method calculates the height of the fiducial marker based on the upper-most and lower-most scan line positions on the shape. The cylinder vertical position error is larger than for the cone. The advantage of the cone fiducial is a more accurate vertical position estimation. When only a few lasers intersect the fiducial marker, the change in arc radius permits estimation of the slope of the cone. This slope indirectly determines the location of the apex and the apex angle. Locating the apex eliminates a degree of freedom for the cone pose along the axis of the cone.

The pose estimates, in Figure 15 and Figure 16, give good precision with a close estimate of the location of a detected fiducial marker. The percentage of correct detections, shown in Table 14, decreases with increased range and the mean pose estimate over frames deviates. At longer ranges, the decreased number of data points from the surface of the fiducial marker gives an increasingly worse fit to the fiducial cross section. This effect is show in Table 15. Incorrect classifications become more frequent in environments where other similar geometric shapes are found. The similar geometric surfaces are erroneously attributed to the fiducial marker.

In summary, cones have better location determination along their vertical axis due to the apex location. Cones are less susceptible to misidentification with natural or man-made shapes, resulting in higher percentages of correct detections and a more accurate mean position estimate. Cylinders on the other hand could easily be located in a mixed structure environment in the form of telephone poles, signposts, structural pillars, or storage tanks. Cylinders show a better range estimate precision than cones.

2. Fiducial Marker Geometry

Initial results showed that the sphere was not large enough to return enough surface points for accurate detection. Therefore, all experiments focused on the cylinder and cone shapes.

The accuracy of the method to detect the cylinder depends on how many laser scan lines intersect the cylinder and how well those intersections are segmented as continuous intervals. The cylinder fiducial marker is intersected by LIDAR scan lines from 10 to 30 meter ranges. The mean number of arcs used to compute the fiducial marker parameters decreases from 7 arcs at 10 meters to 3 arcs at 30 meters.

The cone geometry offers the best means for determining both location and orientation when only a sample of the surface points is available. A sample of the surface points from a cylinder includes ambiguity as to their location along the cylinder axis. A similar sample from a cone or a sphere offers more information to determine the location of the cone apex or sphere center. The cone parameters define a slanted surface that reaches an apex point that is uniquely defined in 3D space. The sphere includes a center point that is unique in 3D. These characteristics of cones and spheres allow a better determination of 3D location than for a cylinder, see Figure 17. The axis line inherent in cylinders and cones provides a means of determining orientation that is not available with a sphere. A sphere has no definable orientation parameter.

The cone fiducial cover did not hold its shape very well as shown in Figure 11. Visual inspection of the point clouds for the 25 meter and 30 meter positions indicate that a continuous interval of data points are segmented along the bottom portions of the cone, but no arcs are fit to the data. This appears to have affected the detection of the lower half of the shape in the LIDAR data.

The constant radius of the cylinder fiducial assists in limiting the number of arcs used to derive the cylinder hypotheses. By using a threshold constraint for the size of arcs used to form cylinder hypotheses, the complexity of the search decreases. The cone fiducial has a range of possible cross section radii along its height and requires that the circular arc radius threshold be opened enough to fit arcs over this range. This increases the computational complexity for finding cones, creating slower detections for cones than cylinders in the same environment. While the method is able to better estimate the position of a cone along its axis, the increased computational complexity may not suit an online processing application. Cylinder fiducial pose estimation is less computationally complex, but leads to position estimate inaccuracies along its axis.

3. Segmentation

Increasing the segmentation threshold leads to under-segmentation of the data, resulting in points from different object surfaces being included in the same line segment. For instance, the area of a cylinder near the base is close to the ground and under-segmentation results in both point data from the ground and the cylinder being sent to feature fitting steps. The increase in outlier data decreases the accuracy of the feature fitting methods. Decreasing the segmentation threshold creates more segments, which separates the points on the cylinder from those that are not on the cylinder. As the segmentation threshold is further decreased, over-segmentation of the same scan line on the cylinder occurs and multiple segments of the same scan line on the same object are detected. With fewer points in each segment, a shorter arc segment is described by the data points and the fit of the corresponding arc geometric feature is worse.

The arc segments near the base of a cone or cylinder fiducial marker are sometimes not detected near the ground. A small segmentation threshold value is required to segment the points on the ground from the points on the fiducial marker. This is especially more true for the cone shape since the transition from the ground to the sloped sides is less abrupt than for a cone. Lowering the segmentation threshold enough segments the points of interest, but also introduces many more segments throughout the data and a corresponding increase in computational complexity. A recommendation is to elevate the marker to provide a space between the marker and surrounding surfaces, creating a sharp gradient and assisting segmentation.

4. Radius Bias

There is an apparent tendency of the method to overestimate the radius of the cylinder with the error increasing with range. This is likely due to the sensor obtaining fewer points of data across a shorter arc length of the fiducial cross section at longer distances. The shorter arcs of point data tend to be flatter and lead to higher arc radius estimates. Also, the beam divergence of the lasers increases the spot size reflecting off of the cylinder surface at longer ranges. This increase in spot size may lead to errors in point data used to find arcs resulting in overestimating the radius.

The number of circular arcs can be minimized with a very low threshold for allowable RMS error that defines a good enough circular arc fit to the segment points. This also limits the number of arcs available to make cylinder and cone shape hypotheses. A higher number of circular arcs increase the complexity of searching for cylinder and cone geometric shape hypotheses. One means to limit the search is to consider sets of circular arcs for fit of a geometric shape if their center points are within a certain range of each other.

5. Detection Accuracy Changes with Range

Problems with fiducial marker tracking include occlusion, misidentification, and noise in the data. Occlusion occurs when another object in the environment lies within the direct line of sight from the sensor viewpoint to the fiducial marker preventing observation of some portion of the marker. Misidentification is either due to false positive or false negative detections of the fiducial marker.

The results depicted in Figure 14 show that the accuracy of fiducial marker detection decreases with range, shown by a falling true positive classification rate. Some possible reasons for this include fewer scan lines crossing the shape and the increase in error of fit with increasing range between the sensor and the fiducial marker. The increase in the arc radius estimate errors introduces more error for the fit of the true marker geometry, such that it is not the method's best choice for a marker hypothesis. Several false positive detections are attributed to trees, garbage cans, and telephone poles with similar radii as the fiducial marker.

B. FUTURE WORK

The true orientation of the fiducial markers was not measured before the experiment. Availability of this information might help determine the pose estimation limits with respect to variations in orientation.

The fiducial detection method may be improved by using an ellipse feature fitting step instead of circular arcs. The intersection of the laser scan lines with the cylinder or cone shapes are not true circular arcs. The intersection is defined as a conic section. An ellipse fitting step might be able to find more cross-sectional arcs at a wider range of fiducial marker orientations. This may permit a more accurate detection of a wider range of fiducial marker poses.

One purpose for using fiducial markers with LIDAR range data is to register the range data with other imaging sensor data. Markers that are identifiable in the image data as well as the range data could then be used for registration. An investigation of markers suited to these dual purposes (such as colored cones) is desirable.

A final interesting idea for future work is to use the cylinder fiducial pose estimation method to detect and estimate the relative pose of telephone poles from a vehicle driving down a street. The effectiveness of the estimation method could be investigated for precision and accuracy of localizing the vehicle in a dynamic street environment, without requiring artificially placed fiducials.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS

LIDAR sensors produce precise range data of the surfaces in the sensed environment. The method presented in this thesis provides a means for detecting cylinder and cone features within the LIDAR data. The accuracy of the range data allows the method to estimate a precise pose for cylinder or cone shapes employed as fiducial markers. The experiments showed that the accuracy of the method to detect cone fiducial markers is better than for cylinder fiducial markers. The accuracy decreases with increasing range for any fiducial shape. At longer ranges, fewer laser scan lines intersect the fiducial marker producing fewer data points for the geometric fitting steps.

For cylinder fiducial pose estimates, this method maintains a relatively small range estimate error with a larger estimate error in the vertical axis location. The vertical axis location estimate error is smaller for the cone, but the range error was greater than for the cylinder.

The geometric parameters of a cylinder or cone fiducial marker provide a means to detect the marker within the point data and to estimate the pose of the fiducial marker relative to the sensor. If the pose of the fiducial marker is known within the environment coordinates, the pose of the LIDAR sensor in the environment can be derived. The cone fiducial marker provides a more accurate means for pose estimation because the apex constrains the position along the cone axis. Cylinder pose estimation is ambiguous along its axis. Cylinders are less computationally expensive to detect than cones. With a more accurate pose estimate along the axis, fiducial markers shaped as cones offer the most potential for data registration, vehicle localization, and applications requiring LIDAR pose estimation without the use of a GPS or an IMU. That said, the method routinely achieved a more precise estimate of fiducial marker range for cylinders than for cones.

For many applications, online processing of the LIDAR data for fiducial detection and pose estimation is ideal. The presented method has the potential to be used with online LIDAR data streams, but has not yet been tested. Detecting the cylinder fiducial is more efficient than the cone when the radius threshold is near the true fiducial radius.

With a close tolerance for radius, there are fewer arcs and axis line hypotheses to test. The tradeoff is between gaining a more accurate pose estimate from a cone and using a less computationally expensive cylinder estimate.

APPENDIX: VELODYNE HDL-64E S2 LIDAR

A. SYSTEM DESCRIPTION

The Velodyne HDL-64E S2 LIDAR System (VLS) is a terrestrial LIDAR system designed for use on ground vehicles. This system was originally designed for environment sensing in the DARPA Grand Challenge autonomous vehicle competition and was later used by several DARPA Urban Challenge teams as an obstacle detector and input to the navigation system. The VLS has also been used for infrastructure mapping. Figure 19 depicts the Velodyne HDL-64E S2 LIDAR head unit. The two visible glass discs are the laser detectors with two bundles of laser emitters on either side.

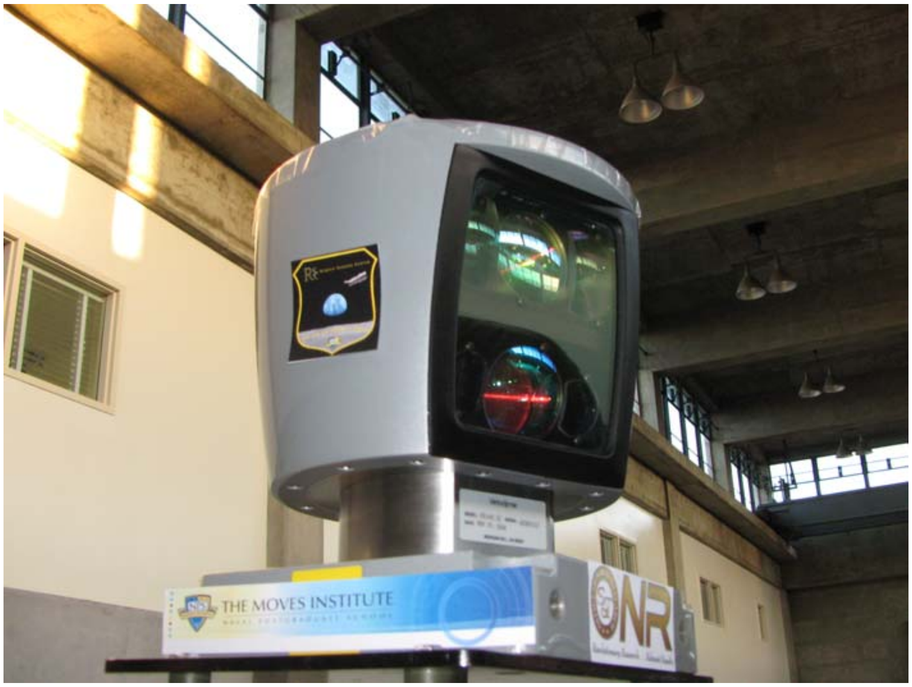


Figure 19. Velodyne HDL-64E S2 LIDAR system.

1. Laser Characteristics

The HDL-64E provides a 360 degree horizontal FOV and a 26.8 degree vertical FOV using 64 pulsed lasers rotating at fixed elevation angles. Rotation speeds can be varied from 300 RPM to 900 RPM. The lasers are eye safe and operate with a 905 nanometer wavelength and a five nanosecond pulse.

The laser firing sequence is determined by the laser position in two blocks of 32 lasers. The order each laser fires inside each block is the same order that the range and intensity data is sent in the network packet. The upper block includes laser numbers [0–31] and the lower block includes laser numbers [32–63].

2. LIDAR Coordinate System Origin and Orientation

The LIDAR coordinate system origin and orientation are used to determine the spatial location of laser reflections. The LIDAR vertical axis is the z-axis, while the 0 and 180 degree rotational positions are on the x-axis, and the 90 and 270 degree positions are along the y-axis. The origin is located in the center of the base plane of the LIDAR. The offset of each laser from this origin position are used to calculate the position of each laser range measurement.

3. Laser Firing Pattern

The lasers are aimed with individual vertical offset angles for vertical coverage of the sensor field of view. Each laser is fired separately with a horizontal offset angle to mitigate cross-talk between laser firings. The HDL-64E S2 generates a high density of points in the horizontal direction of rotation dependent on rotational speed. A higher speed of rotation generates data points more sparsely than a lower rotational speed. Each laser is aimed with a fixed vertical angle, limiting the point density in the vertical direction. The angular resolution is the ability to resolve two objects on adjacent sightlines. Angular resolution is a function of the spatial sampling interval, width of the laser beam, and angular measurement (Cheok, 2005). The angular increment is the angular distance between range samples in either azimuth or elevation. The vertical angular increment is depicted in Figure 20, where

$$D = \Delta d \cong D(1+\tan^2\theta)\Delta\theta$$

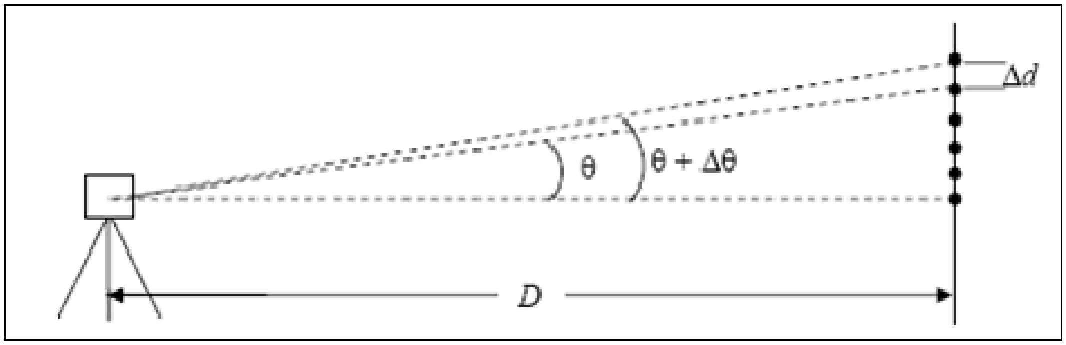
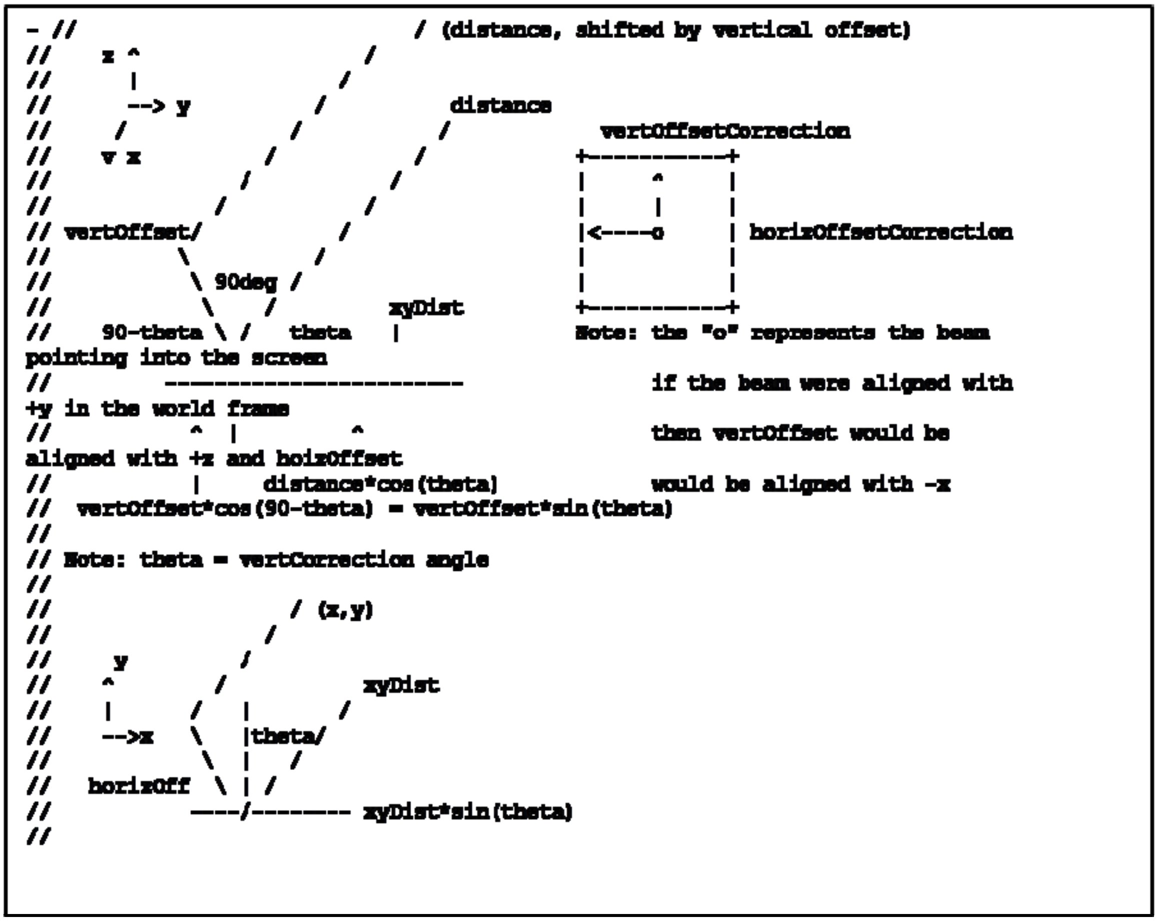


Figure 20. Vertical distance between laser firings (From Cheok, 2005).

To determine the 3-D location of a data point received from the LIDAR system, a rotational and horizontal correction factor must be applied to the range value in the data packet. Factory calibration information is stored in an XML file. A horizontal rotation correction and vertical rotation correction are given in degrees. The range value is corrected for individual laser variance. Finally, vertical and horizontal offset values represent translation from the LIDAR origin. Figure 21 depicts the orientation and direction of the correction values.



```
- //                           / (distance, shifted by vertical offset)
//      z ^                   /
//        |                 /
//        --> y           /          distance
//       /              /          /            vertOffsetCorrection
//      v z           /          /             +-----------+
//                  /          /               |     ^     |
//                /          /                 |     |     |
// vertOffset/             /                   |<----o     | horizOffsetCorrection
//            \          /                     |           |
//             \ 90deg /                       |           |
//              \    /         xyDist          +-----------+
//      90-theta \ /   theta   |               Note: the "o" represents the beam
pointing into the screen
//          ------------------------                 if the beam were aligned with
+y in the world frame
//            ^  |         ^                         then vertOffset would be
aligned with +z and hoizOffset
//            |     distance*cos(theta)              would be aligned with -x
//  vertOffset*cos(90-theta) = vertOffset*sin(theta)
//
// Note: theta = vertCorrection angle
//
//                     / (x,y)
//                   /
//      y          /
//      ^        /           xyDist
//      |      /  |        /
//      -->x   \  |theta/
//              \ |   /
//     horizOff  \ | /
//             ----/-------- xyDist*sin(theta)
//
```

Figure 21. Velodyne LIDAR system laser corrections
(From HDL-64E S2 User's Manual).

The individual laser scan lines with high rotational density and sparse vertical density are visible as concentric arcs along the ground plane of the scene shown in Figure 22. The relatively large gaps between the laser scan lines presents a challenge in processing the data for cross-laser features. This is especially true when considering detection of geometric features, since the relevant features often require several laser scan lines of data to be accurately detected. With sparse vertical information, a geometric shape with a larger vertical dimension is required in order to provide enough laser range returns for accurate geometric feature detection.

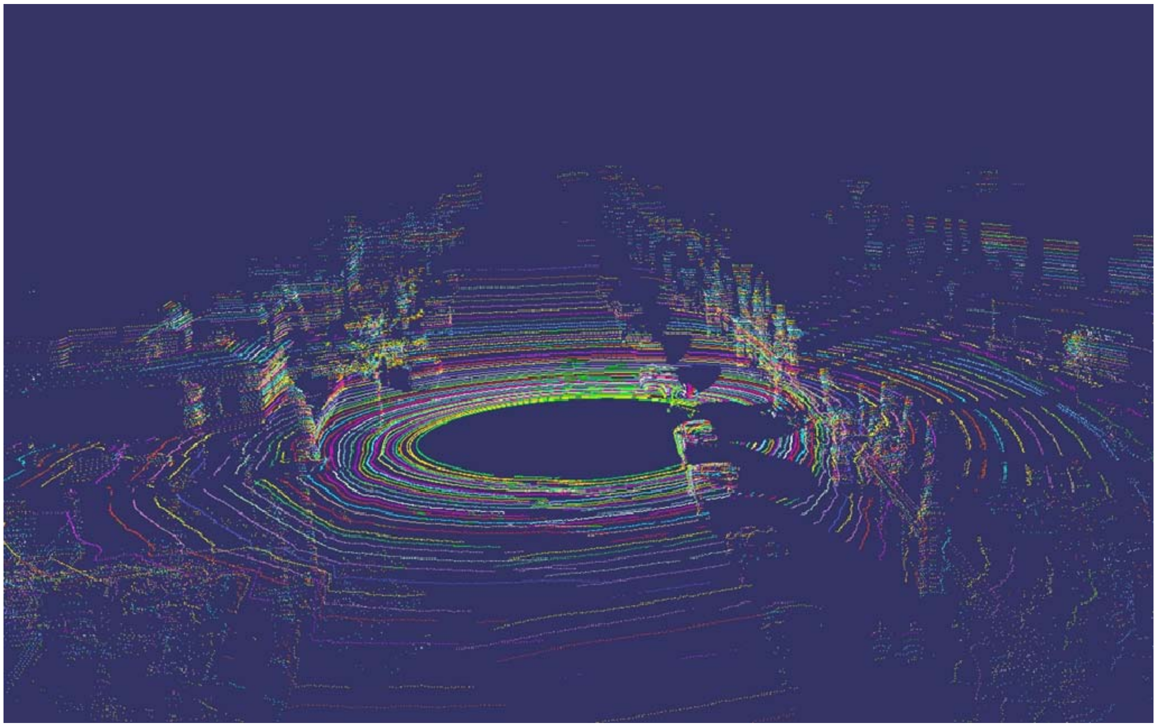


Figure 22. Velodyne LIDAR system laser scan line pattern.

4. Estimated Performance

The manufacturer specified angular resolution is 0.09 degrees in azimuth with estimated distance accuracy of 1.5 centimeters. Tables 19 gives approximate point density and angular resolution values for varying RPM. Tables 20 and 21 show estimated horizontal and vertical resolution estimates based on the rotation speed of the LIDAR.

RPM	RPS (Hz)	Points per revolution (total)	Points Per Revolution per laser	Horizontal Angular Resolution (deg)
300	5	266624	4166	0.0864
600	10	133312	2083	0.1728
900	15	88896	1389	0.2591

Table 19. HDL-64E S2 data sampling resolution estimates.

Distance of Vertical Wall	**10 meters**	**50 meters**	**100 meters**
Vertical Distance between points on the wall	0.0743 m	0.3716 m	0.7430m

Table 20. LIDAR angular resolution.
This table shows the estimated vertical distance between points on a vertically oriented wall at the distance indicated.

Distance of Vertical Wall	**10 meters**	**50 meters**	**100 meters**
Horizontal distance between points (5 Hz)	0.0151 m	0.0754 m	0.1508 m
Horizontal distance between points (10 Hz)	0.0302 m	0.1508 m	0.3016 m
Horizontal distance between points (15 Hz)	0.0452 m	0.2261 m	0.4522 m

Table 21. LIDAR angular resolution.
This table shows the horizontal distance between points on a vertical wall at the distance indicated.

5. Operation

The HDL-64E can be mounted at any angle from zero to 90 degrees. The horizontal configuration is generally used for vehicle sensing in 360 degrees and the side configuration is used to generate data at higher elevations for environments including trees, power lines, and buildings.

The spin rate ranges from 300 RPM (5 Hz) to 900 RPM (15 Hz) and controls the horizontal angular resolution. The spin rate is controlled via a serial command passed over a RS-232 COM port.

6. System Output

The system output is transmitted using the UDP network protocol via a standard RJ-45 Ethernet cable. Each UDP Ethernet packet contains a data payload of 1206 bytes consisting of 12 firing blocks of 100-byte firing data followed by six bytes of device status information. The 64 lasers are divided into an upper and lower firing block of 32 lasers each. One of the 12 firing blocks represents the firing data from either the upper or lower block. Each UDP packet contains six separate firings for each laser that are grouped by firing block. (HDL-64E S2 User's Manual) Table 22 summarizes the byte order of the packet data contents.

Byte Order	**Data Contents**
2 bytes	Header info. Identify upper/lower block.
2 bytes	Rotational info. [0-35999] hundredths of a degree.
3 bytes (x 32)	Laser return info. 32 sequential laser returns with two bytes of range information in 2 mm increments followed by a one byte intensity value [0-255]. No laser return within 120 meters records a zero range value.
6 bytes	Status info. Two byte incremental spin count [0-65,535] and an alternating four byte internal temperature or 4 byte firmware version number.

Table 22. HDL-64E S2 data packet format (After HDL-64E S2 User's Manual).

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Adams, M. D. (1998). *Sensor modelling, design and data processing for autonomous navigation*. New Jersey: World Scientific Publishing Company.

Alda, J. (2003). Laser and gaussian beam propagation and transformation. In *Encyclopaedia of Optical Engineering* (pp. 999–1013). New York: Marcel Dekker.

Anderson, H. E. (2006, July 31). The use of high-resolution remotely sensed data in estimating crown fire behavior variables. Final Report to the Joint Fire Science Program, Project 01-1-4-07. Retrieved on March 2, 2010, from http://jfsp.nifc.gov/JFSP_Completed_Projects_6.htm

ASTM Standard E2544-09b (2009). Standard Terminology for Three-Dimensional (3D) Imaging Systems. West Conshohocken, PA: ASTM International. doi: 10.1520/E2544-09B10.1520/C0033-03

Atieg, A & Watson, G. A. (2004). Fitting circular arcs by orthogonal distance regression. *Applied Numerical Analysis & Computational Mathematics. 1*(1), pp. 66–76. doi: 10.1002/anac.200310006

Bae, K., & Lichti, D. D. (2004). Automated registration of unorganised point clouds from terrestrial laser scanners. *International Archives of Photogrammetry and Remote Sensing (IAPRS*), *35*(B5), 222–227.

Baltzakis, H., Argyros, A., & Trahanias, P. (2003). Fusion of laser and visual data for robot motion planning and collision avoidance. *Machine Vision and Applications*, *15*(2), 92–100. doi: 10.1007/s00138-003-0133-2.

Besl, P. J. (1988). *Surfaces in range image understanding*. New York: Springer-Verlag Inc.

Biggers, K., Keyser, J., & Wall, J. (2009). Automated reconstruction of synthesized environments from complex point cloud datasets. *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2009.* Retrieved on March 2, 2010, from http://ntsa.metapress.com/link.asp?id=y676562h0241t626

Bolles, R. C., & Fischler, M. A. (1981). A RANSAC-based approach to model fitting and its application to finding cylinders in range data. *Proceedings of the 7th International Joint Conference on Artificial Intelligence, vol. 2,* 637–643.

Bolles, R. C. & Fischler, M. A. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Bradski, G., & Kaehler, A. (2008). Learning OpenCV. Cambridge: O'Reilly.

Bullock, D. (2008, January 31). Gallery: Inside the navy's armed-robot labs. *Wired Magazine*, Retrieved on February 22, 2008, from: http://www.wired.com/politics/security/multimedia/2008/01/gallery_spawar?slide=10&slideView=6

Cawood, S. & Fiala, M. (2007). *Augmented reality: A practical guide.* Raleigh, NC: The Pragmatic Bookshelf.

Chaperon, T., & Goulette, F. (2001). Extracting cylinders in full 3D data using a random sampling method and the gaussian image. *Proceedings of the Vision Modeling and Visualization Conference, 2001*, pp. 35–42.

Cheok, G. S., Ed. (2005). *Proceedings of the 2nd NIST LADAR performance evaluation workshop–March 15-16, 2005.* NISTIR 7266. Gaithersburg, MD: National Institute of Standards and Technology.

MATLAB® Central, (n.d). *Circle Fit (Taubin method).* Retrieved on July 17, 2009, from http://www.mathworks.com/matlabcentral/fileexchange/22678

Duda, R. O. & Hart, P.E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the Association of Computing Machinery, 15*(1), pp. 11–15.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification*. New York: Wiley-Interscience.

Elstrom, M. D., Smith, P. W., & Abidi, M. A. (1998, October). Stereo-based registration of LADAR and color imagery. *SPIE Conference on Intelligent Robots and Computer Vision, 3522(343),* 343–354. doi: 10.1117/12.325781

Feddema, John and Little, Charles. (1997, April) Rapid world modeling: Fitting range data to geometric primitives. *IEEE International Conference on Robotics and Automation*, 4(), 2807–2812.. doi: 10.1109/ROBOT.1997.606712

Fitzgibbon, A., & Fisher, R. B. (1995). A buyer's guide to conic fitting. In *British Machine Vision Conference, 2(),* 513–522.

Fitzgibbon, A., Pilou, M., & Fisher, R.B. (1999). Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, 21*(5), 476–480.

Flor, M. D. L. (2003). *The carrara studio 3 handbook*. Charles River Media.

Forsyth, D. A., & Ponce, J. (2003). *Computer vision: A modern approach*. Upper Saddle River, NJ: Pearson Education, Inc.

Gachter, S., Nguyen, V., & Siegwart, R. (2006). Results on range image segmentation for service robots. In *IEEE International Conference on Computer Vision Systems, 2006 ICVS '06.* p. 53. doi: 10.1109/ICVS.2006.54.

Gao, C., Hoffman, I., Miller, T., Panzarella, T., & Spletzer, J. (2008). Autonomous docking of a smart wheelchair for the automated transport and retrieval system (ATRS). *Journal of Field Robotics*, *25(*4-5*),* 203–222.

Gao, C., Hoffman, I., Miller, T., Panzarella, T., & Spletzer, J. (2007). Performance characterization of LIDAR based localization for docking a smart wheelchair system. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), Workshop on Assistive Technologies: Rehabilitation and Assistive Robotics*.

Golub, G.H. and Van Loan, C.F. (1996). *Matrix computations.* Baltimore, MD: Johns Hopkins University Press.

Haas, G. A. (2005). *Wall-based registration of two scanning LADAR sensors*. Aberdeen Proving Ground, MD: Army Research Laboratory.

Hanna, B., Chai, B., Hsu, S. (2005). Wide-area terrain mapping by registration of flash LIDAR imagery. *Proceedings of the SPIE, 5791, pp. 193-207.*

HDL-64E S2 User's Manual. (n.d.). Velodyne, Inc. Retrieved on March 2, 2010, from http://www.velodyne.com/lidar/products/manual/HDL-64E%20S2%20manual_Rev%20B_web.pdf

Hoff, W.A, Nguyen, K., & Lyon, T. (1996). Computer vision-based registration techniques for augmented reality. *Proceedings of Intelligent Robots and Computer Vision XV, SPIE 2904*, pp. 538-548.

Isenburg, M., Liu, Y., & Snoeyink, J. (2006). Streaming extraction of elevation contours from LIDAR points. Retrieved on March 13, 2010, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.79.9669

Kasa, I. (1976). A circle fitting procedure and its error analysis. *IEEE Transactions on Instrument Measurement*, *25*(3)*, 8–14.*

Lalonde, V,N., Huber, D.F., & Hebert, M. (2006) Natural terrain classification using three dimensional LADAR data for ground robot mobility. *Journal of Field Robotics 23(10) 839–861*.

Lindberg, B., Jones, J., & Kolsch, M. (2009). Panoramic augmented reality for persistent information in counterinsurgency environments. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2009*. Retrieved on March 2, 2010, from http://ntsa.metapress.com/index/X82V2M4314435657.pdf

Neumann, U., You, S., Hu, J., Jiang, B., & Lee, J. (2003). Augmented virtual environments (AVE): Dynamic fusion of imagery and 3d models. *Proceedings of the IEEE Virtual Reality*, *2003*, 61–67.

Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., & Siegwart, R. (2007). A comparison of line extraction algorithms using 2D range data for indoor mobile robotics. *Autonomous Robots, 23*(2), 97–111.

Ogundana, O., Coggrave, C., Burguete, R., & Huntley, J. (2007). Fast Hough transform for automated detection of spheres in three-dimensional point clouds. *Optical Engineering 46(5), 051002*.

O'Leary, P., & Zsombor-Murray, P. (2004). Direct and specific least-square fitting of hyperbolae and ellipses. *Journal of Electronic Imaging*, *13*(3), 492–503.

Premebida, C., & Nunes, U. (2005) Segmentation and geometric primtives extraction from 2D laser range data for mobile robot applications. *Robotica 2005 – Actas do Encontro Cientifico.* Retrieved on May 15, 2009, from http://cyberc3.sjtu.edu.cn/doc/paper/ISRRobotica05.Pdf

Rabbani, T., & Van Den Heuvel, F. (2005). Efficient hough transform for automatic detection of cylinders in point clouds. In *Proceedings of the ISPRS Workshop on Laser Scanning, Netherlands, September 12-14, 2005*.

Rusu, C., Tico, M., Kuosmanen, P., & Delp, E. (2003). Classical geometrical approach to circle fitting -review and new developments. *Journal of Electronic Imaging 12(1), 17 –193*.

Schnabel, R., Wahl, R., & Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, *26*(2), 214–226.

Burgard, W., & Hebert, M. (2008). World Modeling. In Siciliano, B., & Khatib, O. (Eds.) *Springer Handbook of Robotics* (pp. 853–869). Berlin: Springer-Verlag.

Shakarji, C. M. (1998). Least-squares fitting algorithms of the NIST algorithm testing system. *Journal of Research National Institute of Standards and Technology, 103*(6), 633–641.

Shirley, P., & Ashikhmin, M. (2005). *Fundamentals of computer graphics*. Wellesely, MA: A K Peters, Ltd.

Steinbis, J., Hoff, W., & Vincent, T. L. (2008). 3D fiducials for scalable AR visual tracking. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (pp. 183–184). doi: 10.1109/ISMAR.2008.4637357

StreetMapper: Mobile Laser Mapping. (n.d.). Retrieved on March 12, 2010, from http://www.streetmapper.net/index.htm

Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2007). Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LIDAR data. IAPRS Volume XXXVI, Part 3 / W52, 2007.

Taubin, G. (1991). Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(11), 1115–1138.

Van der Glas, M., Vos, F. M., Botha, C. P., & Vossepoel, A. M. (2002). Determination of position and radius of ball joints. In *Proceedings of the SPIE International Symposium on Medical Imaging*, *4684*(1571), 1571-1577. doi: 10.1117/12.46712

Wehr, A. (2005). Laser scanning and its potential to support 3D panoramic recording. In *Proceedings of the ISPRS Workshop on Panoramic Photogrammetry*. Retrieved on March 2, 2010, from http://www.isprs.org/proceedings/XXXVI/5-W8/Paper/PanoWS_Berlin2005_Wehr.pdf

Yoon, J., Sagong, M., Lee, J., & Lee, K. (2009). Feature extraction of a concrete tunnel liner from 3D laser scanning data. *NDT & E International*, *42*(2), 97-105. doi: 10.1016/j.ndteint.2008.10.001

Zhao, J. , Roberts P.T.E., Colchester, A.C.F., Holton-Tainter, K. (1996). Registration updating using marker pins in a video based neurosurgical guidance system (VISLAN). Lecture Notes in Computer Science; Vol. 1131. *Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pp. 287–196.

Zhao, W., Nister, D., & Hsu, S. (2005). Alignment of continuous video onto 3D point clouds. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1305–1318.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Mathias Kölsch
   Naval Postgraduate School
   Monterey, California

4. Timothy Chung
   Naval Postgraduate School
   Monterey, California